# NAVAL POSTGRADUATE SCHOOL

## MONTEREY, CALIFORNIA

# THESIS

**WEB-BASED ARMY REPEATABLE LESSON IN OPERATIONAL COMBAT (WARLOC)**

by

Michael K. Pavek
Austin T. Starken

June 2014

Thesis Co-Advisors: Christian Darken
Jeffrey Appleget

**This thesis was performed at the MOVES Institute.**
**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

# REPORT DOCUMENTATION PAGE

*Form Approved OMB No. 0704–0188*

*Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.*

| | | |
|---|---|---|
| 1. AGENCY USE ONLY *(Leave Blank)* | 2. REPORT DATE<br>06-20-2014 | 3. REPORT TYPE AND DATES COVERED<br>Master's Thesis 07-02-2012 to 06-20-2014 |

**4. TITLE AND SUBTITLE**

WEB-BASED ARMY REPEATABLE LESSON IN OPERATIONAL COMBAT (WAR-LOC)

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**

Michael K. Pavek, Austin T. Starken

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Naval Postgraduate School
Monterey, CA 93943

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

N/A

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(maximum 200 words)*

The current generation of junior officers have spent the last ten years continuously deployed with tactical units focused on stability operations for both training and execution. This has led to a significant degradation in Major Combat Operations (MCO) planning and execution abilities at both the tactical and especially the operational-level. Faced with an unclear operating environment and a population of experienced combat vets, the senior military leaders must determine the training direction of the armed services. In a fiscally-constrained environment, web-based learning games can assist in rapidly bridging the gap between the familiar and unfamiliar at a lower cost and at increased adoption rates versus large simulations. This thesis describes the designing and development process of an original web-based wargame that familiarizes junior field-grade and senior company-grade officers on MCO operations at the joint level. The end state was the production of a completed paper wargame and a VASSAL module which adapted the paper version into an electronic form. Lastly, using HTML5, Javascript and mainly free-to-use tools, we created a prototype wargame that only requires a web browser for service members to play.

**14. SUBJECT TERMS**

Major Combat Operations, Joint Doctrine, Game Design, Game-based Learning, HTML5, JavaScript, CSS, Software Engineering

**15. NUMBER OF PAGES** 165

**16. PRICE CODE**

| | | | |
|---|---|---|---|
| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
| Unclassified | Unclassified | Unclassified | UU |

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

# WEB-BASED ARMY REPEATABLE LESSON IN OPERATIONAL COMBAT (WARLOC)

Michael K. Pavek
Lieutenant Colonel, United States Army
B.A., St. John's University, 1996

Austin T. Starken
Captain, United States Army
B.S., Florida Institute of Technology, 2005

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN
MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION**

from the

**NAVAL POSTGRADUATE SCHOOL
June 2014**

Authors: Michael K. Pavek

Austin T. Starken

Approved by: Christian Darken
Thesis Co-Advisor

Jeffrey Appleget
Thesis Co-Advisor

Christian Darken
Chair, MOVES Academic Committee

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

The current generation of junior officers have spent the last ten years continuously deployed with tactical units focused on stability operations for both training and execution. This has led to a significant degradation in Major Combat Operations (MCO) planning and execution abilities at both the tactical and especially the operational-level. Faced with an unclear operating environment and a population of experienced combat vets, the senior military leaders must determine the training direction of the armed services. In a fiscally-constrained environment, web-based learning games can assist in rapidly bridging the gap between the familiar and unfamiliar at a lower cost and at increased adoption rates versus large simulations. This thesis describes the designing and development process of an original web-based wargame that familiarizes junior field-grade and senior company-grade officers on MCO operations at the joint level. The end state was the production of a completed paper wargame and a VASSAL module which adapted the paper version into an electronic form. Lastly, using HTML5, Javascript and mainly free-to-use tools, we created a prototype wargame that only requires a web browser for service members to play.

THIS PAGE INTENTIONALLY LEFT BLANK

# Contents

**Appendices**

# List of Figures

# List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

# List of Acronyms and Abbreviations

| | |
|---|---|
| **AAA** | antiaircraft artillery |
| **ADA** | air defene artillery |
| **AE** | attacker eliminated |
| **API** | application programming interface |
| **AR** | attacker retreat |
| **ASM** | air-to-surface missiles |
| **ATO** | air tasking order |
| **BM** | ballistic missiles |
| **BotB** | *Battle of the Bulge* |
| **CAS** | close air support |
| **C2** | command and control |
| **CCDR** | combatant commander |
| **CFACC** | Combined Force Air Component Command |
| **CFC** | Combined Forces Command, Korea |
| **CFLCC** | Combined Forces Land Component Command |
| **CJTF** | combined joint task force |
| **CIO** | Chief Information Officer |
| **CM** | cruise missile |
| **COCOM** | combatant command |
| **COG** | center of gravity |

| | |
|---|---|
| **COIN** | counterinsurgency |
| **CoN** | certificate of networthiness |
| **CRT** | combat resolution table |
| **CSS** | Cascading Style Sheets |
| **DCA** | defensive counterair |
| **DE** | defender eliminated |
| **DMZ** | demilitarized zone |
| **DNBI** | disease non-battle injuries |
| **DoD** | Department of Defense |
| **DR** | defender retreat |
| **DRM** | dice roll modifier |
| **DT** | disrupted |
| **EAB** | echelons above brigade |
| **EAD** | echelons above division |
| **EE** | equal elimination |
| **FEBA** | forward edge battlefield area |
| **GDD** | game design document |
| **GDW** | Game Designer’s Workshop |
| **GIMP** | GNU Image Manipulation Program |
| **HTML** | HyperText Markup Language |
| **HSC** | Homeland Defense Security Council |

| | |
|---|---|
| **HVAA** | high value airborne assets |
| **IADS** | integrated air defense systems |
| **IA** | Information Assurance |
| **IDE** | integrated development environment |
| **IE** | Internet Explorer |
| **IRBM** | intermediate-range ballistic missile |
| **ISR** | intelligence, surveillance, and reconnaissance |
| **JFC** | joint force commander |
| **JTF** | joint task force |
| **JP** | Joint Publication |
| **KPA** | Korean People's Army |
| **KPAF** | Korean People's Army Air Force |
| **KPAGF** | Korean People's Ground Force |
| **LOC** | line of communications |
| **MRBM** | medium-range ballistic missile |
| **MCO** | major combat operations |
| **MSC** | major subordinate command |
| **NE** | no effect |
| **NEC** | Network Enterprise Center |
| **NPS** | Naval Postgraduate School |
| **NSC** | National Security Council |

| | |
|---|---|
| **OCA** | offensive counterair |
| **OS** | operating system |
| **OSS** | Open Source Systems |
| **PBEM** | play by email |
| **PC** | personal computer |
| **PLA** | People’s Liberation Army |
| **PLAAF** | People’s Liberation Army Air Force |
| **PLAGF** | People’s Liberation Army Ground Force |
| **POTUS** | President of the United States |
| **PRC** | People’s Republic of China |
| **ROK** | Republic of Korea |
| **ROK-US** | Republic of Korea - United States |
| **SAM** | surface-to-air missile |
| **SEAD** | suppression of enemy air defenses |
| **SRF** | strategic rocket forces |
| **SecDef** | Secretary of Defense |
| **SOF** | special operations forces |
| **SPI** | Simulations Publications Incorporated |
| **SRBM** | short-range ballistic missile |
| **TBM** | theater ballistic missile |
| **TOC** | Total Ownership Cost |

| | |
|---|---|
| **UN** | United Nations |
| **U.S.** | United States |
| **WARLOC** | Web-based Army Repeatable Lesson in Operational Combat |
| **WWI** | World War I |
| **WWII** | World War II |
| **ZOC** | zone of control |

THIS PAGE INTENTIONALLY LEFT BLANK

# Acknowledgements

We would like to thank a number of people that truly made this thesis possible. First and foremost, we would like to thank our wives for their patience, understanding, and unwavering support during this entire process. Often times they would pull much more than their share of work to keep the households running and we are eternally grateful. We would like to thank Melissa specifically for her wonderful artwork on WARLOC's logo.

Next we would like to thank our advisors, Dr. Chris Darken and COL(R) Jeff Appleget, for their outstanding inspiration and guidance. In particular, we would like to thank Dr. Darken for his willingness to step outside the realm of the typical thesis topics and delve into a very hands-on exploration of game-based learning.

Lastly, we would like to thank all the commercial wargame designers/developers that assisted us. This includes Roger Miller from Revolution Games, Joe Miranda from One Small Step games, Alan Emrich from Victory Point Games, and Joel Uckelman from the VASSAL Development Team. We would particularly like to thank Gary Morgan from Up and Running and John Tiller from John Tiller Software for their in-depth advice and assistance. These two designers were absolutely critical to the design and development of our wargame and we can't thank them enough for their perspective and scoping of our thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 1:
# Introduction

## 1.1 Background

The United States military is at a critical if unrecognized nexus point. The continual focus on stability operations has weakened leaders' skills in major combat operations (MCO). After more than a decade of counterinsurgency (COIN) operations in Iraq and Afghanistan, the United States military has become extremely proficient in tactical stability operations. The consequence of this continued focus on the tactical level of war is that little time is available for Army officers to train on MCO. Unsurprisingly, some junior field-grade and most company-grade officers are lacking skills in MCO [1]. Though these skills are still present within senior field grade officers and Flag Officers/General Officers, they will soon disappear as these groups retire from the service.

Organizationally, the United States (U.S.) Joint Forces, and particularly the U.S. Army, currently focus on their ability to lead and support stability operations. The 1991 collapse of the Soviet Union shifted the future operating environment from large-scale regional conflicts to small-scale conflicts that include a greater focus on stability operations. In reaction to this change, the Army conducted an extensive force structure transformation that changed the organizational structure from division-centric to brigade-centric [1]. The total number of brigades in the U.S. Army increased, as did their individual capabilities which resulted in fully manned brigades and hollowed echelons above brigade (EAB) units. Today, junior officers spend the majority of their time at brigade or below thus receiving little to no MCO training [1]. Consequently, officers lack familiarity with both operational-level commands and MCO at a point in their careers when these very skills are crucial. Upon assignment to corps, field armies, or joint commands, these officers struggle to execute tasks that were once considered rudimentary [1]. An officer corps incapable of such tasks is a significant strategic vulnerability within the U.S. military.

Opportunities to prepare officers for operational-level MCO, prior to their arrival at higher-level units, are fleeting. The U.S. Army is just beginning to shift focus back to MCO warfighting exercises [2]. Currently, senior leaders must spend considerable time instructing their junior officers on basic MCO skills that, in the past, would have developed naturally through on the job experience. This additional requirement of training basic skills diminishes the amount of advanced training covered within any given exercise.

### 1.1.1 Wargames

Peter Perla defines a wargame as "a warfare model or simulation in which the flow of events shapes, and is shaped by, decisions made by a human player or players during the course of those events" [3]. Wargames come in two forms, the professional version used to train service members of the armed forces and the commercial versions used by hobbyists [4]. These wargames span conflicts across human history with a scale that ranges from small unit missions to grand sweeping global conflicts. A commanding officer could easily access this vast breadth of topics and create a library of scenarios to train his officers on MCO in a modern or near-future conflict.

The two primary formats of wargames are either the paper tabletop format or the computerized format. Several examples of operational-level wargames exist within the paper and computer wargaming community. Paper wargame examples include *Next War Korea* (GMT Games), *Corps Command* (Lock N' Load Publishing), *NATO: The Next War in Europe* (Victory Games), *Central America* (Victory Games), and *Gulf Strike* (Victory Games). Computer wargame examples include the *Operational Art of War* (Matrix Games), *Battle of the Bulge* (Shenandoah Studio), and *Panzer Campaigns* (John Tiller Software). These wargames provide useful insight on how to model a joint task force (JTF) MCO scenario.

Paper wargames are often very detailed and can be very complex. Game play for a single game from start to finish can last anywhere from 30 minutes to multiple days based on their complexity. Computerized wargames help mitigate this complexity by automating many tasks that are typically conducted by the players. Computerized wargames require players to purchase and install licensed copies of the games on their local computer systems. The complexity of learning all the rules of a paper wargame or the time required to locate, install, and configure a current licensed copy of a computerized wargame requires a substantial amount of effort before casting the first die.

Constant deployments, day to day training, administrative requirements and family time all compete for a typical military officer's attention. It is often difficult for officers to set aside time to focus on their personal, professional development. Though paper and commercial wargames can be entertaining while providing valuable insight into abstract concepts, they are time consuming. Making wargames easy to access and easy to learn significantly reduces the game's time requirement. Using a game, rather than doctrinal text, engages players to use their problem solving skills that may result in experiential learning [5].

### 1.1.2 Web Development

Moore's Law suggests that the average computing power doubles every two years [6]. With the advent of the Internet, web technologies are also rapidly improving. Web browsers act as interpreters for various programming languages to include HyperText Markup Language (HTML)5 and Javascript. Anyone interested in writing an application using HTML5 and Javascript only needs a basic text editor, like Notepad, and a web browser such as Microsoft Internet Explorer (IE) or Google Chrome. Internet access is required to provide client computers access to the application once developed and published to a server. Interestingly enough, Notepad and IE come bundled in the Windows operating system (OS), which is the standard OS used by the Army.

HTML5 is a direct descendent of HTML and has always been relatively easy to grasp when compared to more complex languages like C++. Javascript, initially intended to enhance the functionality of web pages written in HTML can now be used to create three-dimensional virtual worlds in the Internet browser of almost any computer [7]. Using the web browser as an interpreter removes the requirement to compile code into machine language. Since a web browser is used to execute the code it can also be used to test it. If the code works on the programmer's local machine using a specific web browser, then it will execute on any client's computer using the same browser. In fact, due to inherent security restrictions within browsers, it is easier for the code to execute on another client's computer if the executable code itself is stored and accessed by the client from a remote server [7].

With a lower bar for entry and using web browsers to execute code on almost any system, HTML5 and Javascript have rapidly grown in acceptance and use. Javascript receives great support from companies like Google, who offer a host of development tools within their free Internet browser Chrome. In developing the web-based wargame supporting this thesis, Chrome was the browser of choice. Javascript is so accessible that anyone unfamiliar with the language can visit www.khanacademy.com, a free education website currently teaching basic programming through Javascript [8].

## 1.2 Research Problem

The current generation of junior officers has spent the last ten years in tactical units focusing on stability operations in support of continuous deployments. This focus has led to a significant degradation in MCO planning abilities at the operational-level [1]. The U.S. Army understands that future learning models "must seize opportunities to use technology as an enabler to engage and appeal to digital age learners" [9, p. 15]. One possible option in bridging this gap is through the use of a web-based wargame that introduces officers to major combat operations at the operational-level. This wargame would place the user in the command of a JTF conducting MCO against an enemy of equal caliber and strength providing the benefit of simultaneously removing the officer from the tactical-level environment and introducing them to MCO. In addition, it is not overly difficult to create easily accessible web-based wargames at a low cost using freely available open source tools.

## 1.3 Motivation of Research

It is easy for the U.S. Army to look inside and identify learning and training deficiencies. What is difficult is developing low cost solutions to correct the identified deficiencies. With fiscal uncertainty, sequestration, and budget cuts, the U.S. Army cannot easily allocate funding to resolve every shortcoming it finds within its ranks. Often, a unit's mission success results on the ingenuity of one young Soldier who creatively develops a solution to a complex problem.

When it comes to software, software developer contracts often result in the production of turnkey systems. A turnkey system can be used by the Army to fulfill its current capability gaps. However, the inner workings of the system, such as source code, may or may not be accessible to the Army.

As time progresses, threats change, requirements change, and the Army needs updated software. The unfortunate reality is, more often than not, the current software requires only a slight modification to address the Army's newly developed needs. Often, contracts only require contractors to provide an end product with a particular functionality and often leaves the contractor with sole proprietary right to the software's source code. If the Army grows too dependent on the software, "Lock-In" can occur, resulting in the Army's dependence on a single vendor [10]. This dependence makes it difficult for the Army to foster competition to fulfill requirements to update the product once it becomes outdated. The vendor owns the source code making it nearly impossible for other vendors to update it. One possible solution the Army could explore is to develop and thus own their source code. This thesis explores the use of core HTML5 and Javascript code to develop a web-based military wargame to familiarize military officers with basic MCO concepts.

## 1.4 Research Approach

### 1.4.1 Scope

This thesis captures the attempt to create a high quality, easily accessible web-based wargame prototype designed to familiarize junior field-grade and senior company-grade officers on echelons above division (EAD) MCO operations. The thesis end state is a prototype that focuses on land-based JTF operations with the possibility of further development into a complete web-based wargame. The minimum requirement for completion is a turn-based two-player version played on a single computer. A two player networked version is outside the scope of this thesis. A single player version, requiring an artificially intelligent computer opponent, is outside the scope of this thesis as well.

### 1.4.2 Research Questions

1. How can a wargame be designed that introduces operational-level command to a training audience without previous experience in strategic/operational units? What type of learning objectives best support familiarization of operational-level MCO?
2. How can learning objectives be achieved through the use of a web-based wargame?
3. To what degree can a wargame be developed using HTML5 and primarily open source tools that are compatible with both high-end and low-end platforms?

## 1.5 Methodology

The thesis was conducted using the spiral model initially defined by Barry Boehm [11] allowing for an iterative development of the application. Using a spiral methodology provides a number of advantages during the development process. Each cycle consists of a minimum of four phases.

1. Research. The unique requirements of the training audience and which of those requirements that a wargame could potentially meet were identified. Joint and service-specific doctrine, historical examples of real-world missions, and related training methods were researched. Game design, JavaScript programming, and modeling techniques were also extensively researched.
2. Design. The content, rules, learning objectives, scenario, and game mechanics of the wargame were created during this phase. Initial development of the web-based prototype system architecture, game framework, major methods/functions, user interface, and game controls also began to formulate during this phase.
3. Manual Playtesting. During this phase, the thesis team initially produced a paper version of the game to test for play-ability and design flaws to ensure that the game's execution matched the initial vision. Paper playtesting migrated to digital playtesting during this phase through the use of VASSAL, a digital board game development engine. The end state of this phase resulted in the creation of a game design document (GDD) in the form of a VASSAL module.
4. Coding. During this phase, the VASSAL module provided initial insight into the look and feel of a digitized wargame. The many limitations of the VASSAL module helped the thesis team to visualize and thus prioritize the web-based prototype's development. The prototype's current code base was completely written during this phase.

# CHAPTER 2:
# Playing War

An essential element in preparing officers for future strategic/operational assignments is a firm grasp on the theory and the fundamental principles of employing the Armed Forces of the United States. A wargame can provide insight into both the strategic and operational levels of war. Paper wargames are similar to traditional board games in which they use a game board, usually a map, but often have numerous counters to represent individual units. Games in this format can be complex and carry a hefty price tag leading to a limited, or niche, audience [12]. Developing a web-based wargame using open source tools that run in almost any computer's web browser can be made at a negligible cost. Offering the game for a small price, if not free, and hosting it on the web increases the game's accessibility to a larger audience. This chapter summarizes the primary joint theory that is the basis for the wargame design and also reviews several tools that a wargame designer can utilize in the design and development process.

## 2.1 Understanding Operational Level Command

The transition from serving in a tactical command to a joint operational command is a considerable change for many junior field grade officers. It requires a dramatic increase in the scope and scale of operations as well as a significant increase in the number of enablers that support the command. Officers must increase their understanding of MCO operational concepts not just within their own service branch, but also in several other service branches in which they may or may not have any prior experience. They must synthesize a greater amount of information and apply it in an unfamiliar operational setting. These officers require a firm understanding of applied joint theory to succeed. Unfortunately, many of these officers arrive at an operational unit without this foundation [1]. This section summarizes the essential joint theory and relevant doctrine recommended for serving in a joint operational command and forms the basis for the wargame design. It does not attempt to enter a doctrinal discussion of the merits or interpretations of joint and service-specific doctrine.

### 2.1.1 Joint Theory

An essential element in preparing officers for future strategic/operational assignments is a firm grasp on the theory and fundamental principles of employing the Armed Forces of the United States. According to Joint Publication (JP) 1, *Doctrine for the Armed Forces of the United States* [13], “joint” is any activity that involves elements from two or more Military Departments. JP1 coins the term “jointness” and defines it as implying “cross-Service combination wherein the capability of the joint force is understood to be synergistic, with the sum greater than its parts” [13]. In order to achieve this capability, a joint force requires: a high degree of interoperability; joint interdependence; and a commander with the operational authority to assemble a joint force suited for the current mission. Joint interdependence is defined, ”the purposeful reliance by one Service on another Service’s capabilities to maximize complementary and reinforcing effects of both (e.g., synergy)” [13, p. I-2].

In essence, a joint force is one that capitalizes on each other’s strengths, mitigates each other’s weaknesses, understands each other’s Service-oriented language and procedures, and share a common understanding of the operational picture and the mission at hand. The joint force commander (JFC) must assemble a combination of forces that achieve this vision while economizing the size to ensure flexibility and responsiveness. In order to successfully plan and execute joint military operations, leaders must have a basic understanding of the definition of war and its principles. Successful planning and execution of joint military operations can be particularly challenging for leaders that have limited to no experience outside of their own service branch.

The Oxford dictionary defines war as “a state of armed conflict between different nations or states or different groups within a nation or state” [14] while Clausewitz, in his book *On War*, defines war as “an act of violence to compel our opponent to fulfill our will” [15, p. 83]. JP 1 offers, “war is socially sanctioned violence to achieve a political purpose” [13, p. I-3].

#### Warfare

Warfare is how war is waged, it is “the mechanism, method, or modality of armed conflict against an enemy” [13, p. I-4]. Joint doctrine defines the two basic forms of warfare as traditional and irregular [13]. Each of these forms support a fundamentally different strategic purpose, although overlaps will always exist due to the fluid nature of war.

Traditional warfare is defined as “a violent struggle for domination between nation-states or coalitions and alliances of nation-states” [13, p. I-5] with a strategic purpose of the “imposition of a nation’s will on its adversary nation-state(s) and the avoidance of its will being imposed upon us” [13, p. I-5]. Typically nation-states conduct traditional warfare and attempt to influence each other by destroying or defeating the opposing side’s military force or ability to conduct war. This combat occurs throughout all physical domains to include air, land, maritime and also through the information domain, which includes cyberspace. While there is a great deal of emphasis on an opposing nation-state, it should be noted that non-state actors can also conduct traditional warfare operations (e.g., the Viet Cong during the Tet Offensive of 1968).

### Levels of Warfare

Warfare is commonly divided into three levels —strategic, operational, and tactical— which connect actions to objectives as depicted in Figure 2.1. These levels are guidelines but not definitive in their boundaries and it is common to see a great deal of overlaps at the seams of each level. Additionally, due in part to the modern information environment and high level of connectivity between communities, tactical actions can rapidly have a strategic effect.

The President of the United States (POTUS), National Security Council (NSC), and the Homeland Defense Security Council (HSC) as the National Security Staff define national objectives which serve as the focus for the strategic level of warfare [13, p. I-8]. The Secretary of Defense (SecDef) uses the national strategic objectives to develop strategic military objectives which support the military end state and assist the combatant commanders (CCDRs) with theater strategic planning. The CCDRs link national strategy and operational objectives [13, p.I-8].

Operational level planning develops “objectives needed to achieve the military end states strategic objectives” [13, p. I-8]. The CCDRs plan and execute operations using operational art, which according to doctrine is “the use of creative thinking by commanders and staff to design strategies, campaigns, and major operations and organize and employ military forces” [16, p. II-3]. Operational art guides the flow of forces and also “the arrangement

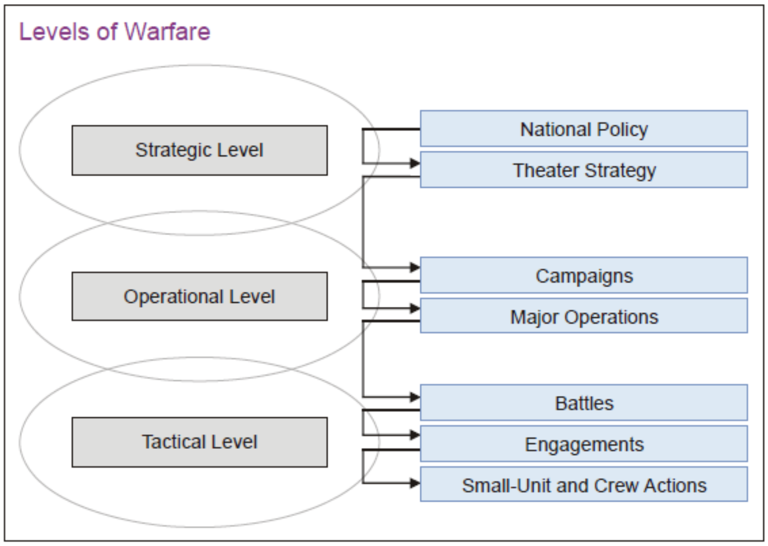


Figure 2.1: Levels of Warfare, from [13, p. I-7]

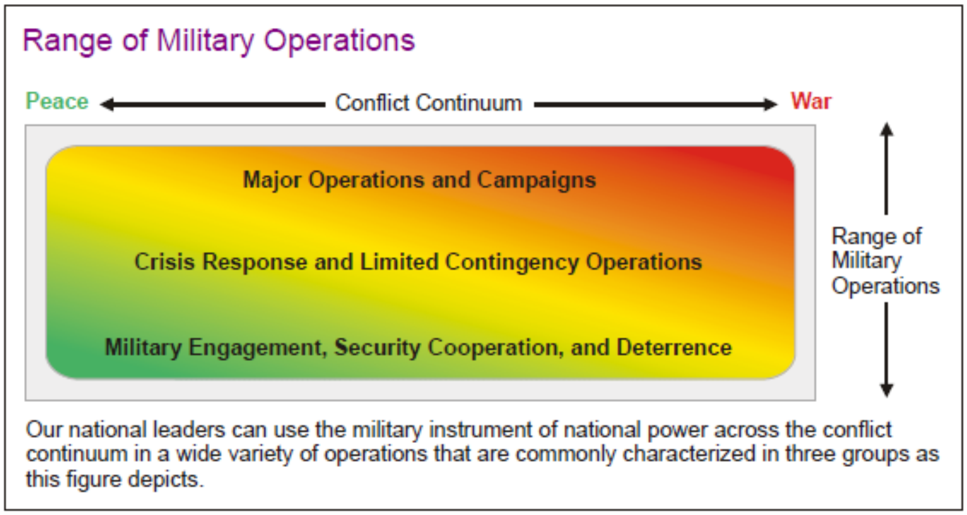
of battles and major operations which support operational and strategic objectives" [13, p. I-8]. Battles, are a set of related engagements involving larger forces than used in engagements and normally affect the course of an operation or campaign [13, p. I-8]. "A campaign is a series of related major operations aimed at achieving strategic and operational objectives within a given time and space" [13, p. I-9] . Campaigns are typically extensive joint operations.

Tactics, are "the employment and ordered arrangement of forces in relation to each other" [13, p. I-8]. It is the level where tactical commanders plan and execute battles and engagements to support combat objectives [13, p. I-8]. Engagements are a tactical conflict, usually between opposing lower echelons maneuver forces [16, p. I-8]. Engagements are typically short-duration events. Operations, are a sequence of tactical actions with a common purpose or unifying theme [13, p. I-9]. Operations may or may not include combat, as well as its support activities such as movement, supply, and maneuver. A major operation combines battles, engagements and strikes to support strategic/operational objectives.

### Range of Military Operations

Joint doctrine categorizes national power into four categories — diplomatic, informational, military, and economic. The U.S. Government leverages these instruments to execute national policy. While the Department of Defense (DoD) is the lead for the military instrument, close coordination and synchronization is necessary across all four instruments of national power [13].

The military instrument can be used in a number of different methods depending on the national strategic objectives and the conflict continuum which ranges from peace to war. Joint doctrine divides these methods into three categories as depicted in Figure 2.2.


Range of Military Operations
Peace
Conflict Continuum
War
Major Operations and Campaigns
Crisis Response and Limited Contingency Operations
Military Engagement, Security Cooperation, and Deterrence
Range of Military Operations

Our national leaders can use the military instrument of national power across the conflict continuum in a wide variety of operations that are commonly characterized in three groups as this figure depicts.

Figure 2.2: Range of Military Operations, from [13, I-14]

Military engagement, security cooperation, and deterrence are activities focused primarily on shaping and maintaining relationships with other nation-states. Crisis response and limited contingency operations are often isolated, small military activities with a defined start and finish or can be part of a larger ongoing mission during combat. Major operations and campaigns aim to rapidly defeat the enemy, eliminate opposition, and develop terms that are beneficial to the U.S. and its inter organizational partners. Major operations and campaigns can be conducted with any mix of offensive, defensive and stability operations [13, p. I-16].

### 2.1.2 Joint Command Organization and Command Relationships

One of the greatest advantages of the U.S. Armed Forces is the wide assortment of capabilities available within the force. These capabilities range across the various levels of war and provide the U.S. decision maker with numerous tools to use within the military instrument of national power. These forces are divided into service branches (Army, Marine Corps, Navy, and Air Force). JP 1 identifies the three levels of organization for joint forces as unified combatant commands, subordinate unified commands, and JTF [13, p. I-15].

A JFC organizes these forces to accomplish their assigned mission and facilitate the joint principles of unity of command and unity of effort. The JFC establishes both service components and functional components as necessary. Figure 2.3 shows some possible components within a joint force.

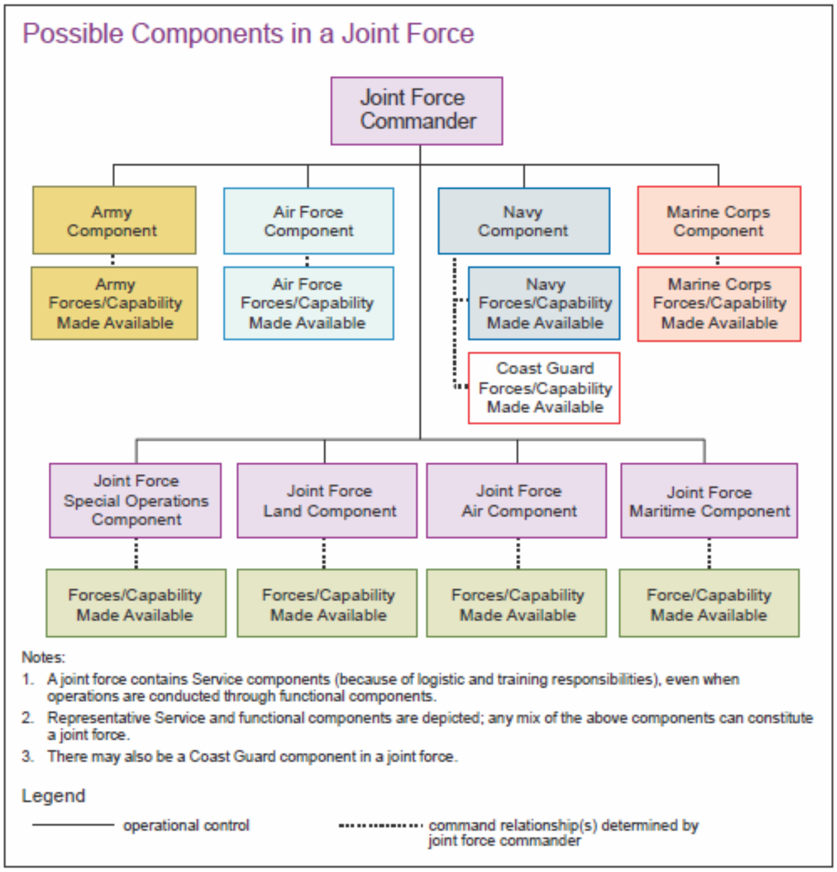

Possible Components in a Joint Force
Joint Force Commander
Army Component
Air Force Component
Navy Component
Marine Corps Component
Army Forces/Capability Made Available
Air Force Forces/Capability Made Available
Navy Forces/Capability Made Available
Marine Corps Forces/Capability Made Available
Coast Guard Forces/Capability Made Available
Joint Force Special Operations Component
Joint Force Land Component
Joint Force Air Component
Joint Force Maritime Component
Forces/Capability Made Available
Forces/Capability Made Available
Forces/Capability Made Available
Force/Capability Made Available
Notes:
1. A joint force contains Service components (because of logistic and training responsibilities), even when operations are conducted through functional components.
2. Representative Service and functional components are depicted; any mix of the above components can constitute a joint force.
3. There may also be a Coast Guard component in a joint force.
Legend
operational control
command relationship(s) determined by joint force commander

Figure 2.3: Possible Joint Force Components, from [13, IV-3]

### Unified Combatant Command

A unified and subunified combatant command (COCOM) are commands with,

> broad continuing missions under a single commander and composed of significant assigned components of two or more Military Departments that is established and so designated by the President through SecDef and with the advice and assistance of the CJCS [13, p. II-11].

A commander of a unified COCOM has several organizational command options as depicted in Figure 2.4.

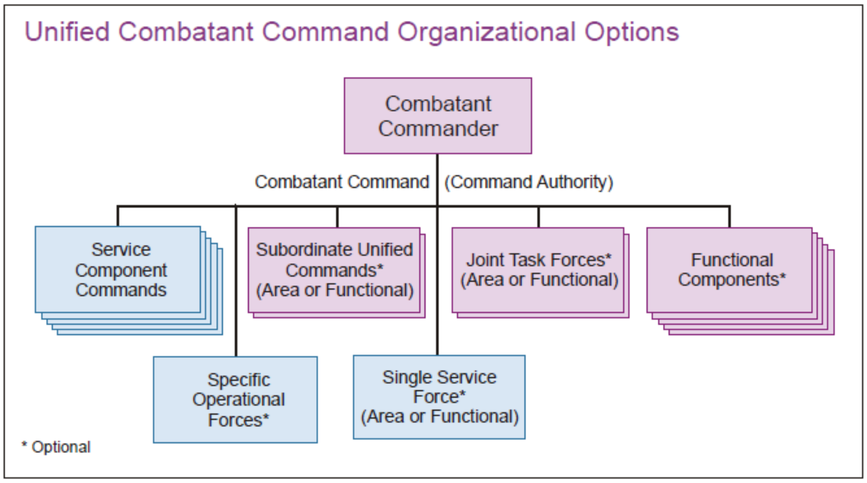

Unified Combatant Command Organizational Options
Combatant Commander
Combatant Command
(Command Authority)
Service Component Commands
Subordinate Unified Commands* (Area or Functional)
Joint Task Forces* (Area or Functional)
Functional Components*
Specific Operational Forces*
Single Service Force* (Area or Functional)
* Optional

Figure 2.4: Unified Combatant Command Organizational Options, from [13, IV-6]

Unified combatant commands are most frequently organized on a geographic basis (known as a geographic combatant command) while subunified commands are established on either a geographical or functional basis.

### Joint Task Force

*JP 1-02 Department of Defense Dictionary of Military and Associated Terms* defines a JTF as "a joint force that is constituted and so designated by the Secretary of Defense, a combatant commander, a subunified commander, or an existing JTF commander" [17, p. 151] Similar to a unified or subunified command, it contains components from the service branches and additional JTFs as depicted in Figure 2.5. The establishing authority assigns the JTF a mission with specific limited objectives.

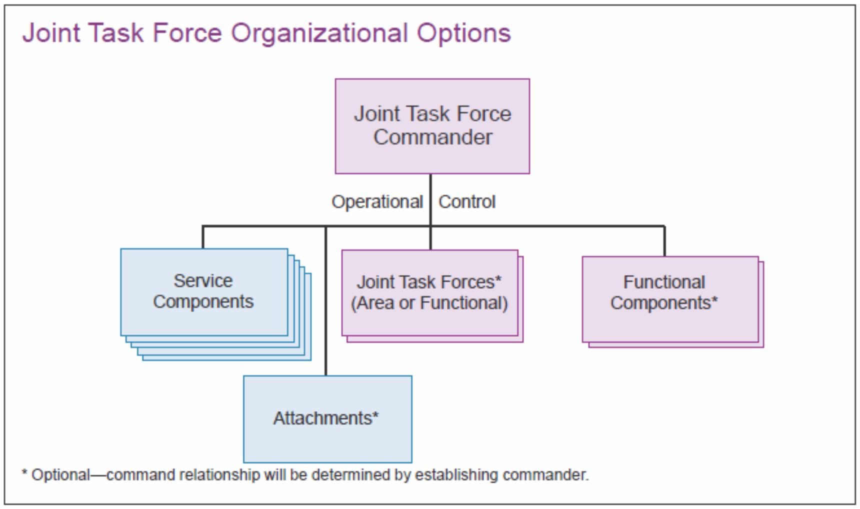

Joint Task Force Organizational Options
Joint Task Force Commander
Operational Control
Service Components
Joint Task Forces* (Area or Functional)
Functional Components*
Attachments*
* Optional—command relationship will be determined by establishing commander.

Figure 2.5: JTF Organizational Options, from [13, p. IV-11]

## 2.2 Joint Functions

A JFC possesses or has rapid access to an entire host of capabilities and resources. The ability of an organization to synthesize and coordinate these diverse activities are often the decisive difference between success and failure. According to *Joint Publication 3-0 Joint Operations*, joint functions are "related capabilities and activities grouped together to help Joint Force Commanders integrate, synchronize, and direct joint operations" [16, p. III-1]. The six common functions that apply to all levels of war entail: "command and control (C2), intelligence, fires, movement and maneuver, protection, and sustainment" [16, p. xiv]. These functions work in conjunction with one another to support the JFC.

### 2.2.1 Command and Control (C2)

C2 is defined as "the exercise of authority and direction by a commander over assigned and attached forces to accomplish the mission" [16, p. III-2]. This entails an extensive number of tasks that are listed in JP 3-0, Chapter 3 to include but are not limited to:

1. Commanding subordinate forces
2. Preparing and modify plans/orders/guidance
3. Assessing the progress toward accomplishing tasks, creating conditions, achieving objectives,

4. Coordinating and controlling the employment of joint lethal and non-lethal capabilities [16, p. III-2].

### 2.2.2 Intelligence

In simplest terms, intelligence can be defined as understanding one's environment. Specifically the intelligence functions "supports this understanding by providing integrated, evaluated, analyzed, and interpreted information concerning foreign nations, hostile or potentially hostile forces or elements, or areas of actual or potential operation" [16, p. III-20]. It tells the commander what the enemy is doing now, what the enemy is capable of doing, and what the enemy might do in the future (most likely and most dangerous courses of action).

### 2.2.3 Fires

The JFC leverages fires to achieve desired effects against the enemy's center of gravity (COG), critical factors, and decisive points. Examples of fires effects include deny, disrupt, delay, suppress, neutralize, destroy, and influence.

Maneuver and joint fire support complement each other in support of the JFC objectives. Effective fires increases freedom of maneuver through the destruction, neutralization, or suppression of enemy forces. Additionally, effective combination of maneuver and fire can place an enemy in a position of either being flanked by maneuvering forces or exposing themselves to joint fires.

A JFC uses weapons and other systems to have a lethal/non-lethal effect on a target. These effects can range from destruction and suppression (lethal) to influencing a population center or key leader. The primary fire support task includes "joint fires that assist air, land, maritime, and SOF to move, maneuver, and control territory, populations, airspace, and key waters" [16, p. III-22]. Additional fires tasks are further discussed in this section.

**Joint Targeting**

Targeting is "the process of selecting and prioritizing targets and matching the appropriate response to them, considering operational requirements and capabilities" [18, p. I-1]. It integrates and synchronizes planning, operations, and intelligence into a joint targeting cycle into six phases as depicted in Figure 2.6.

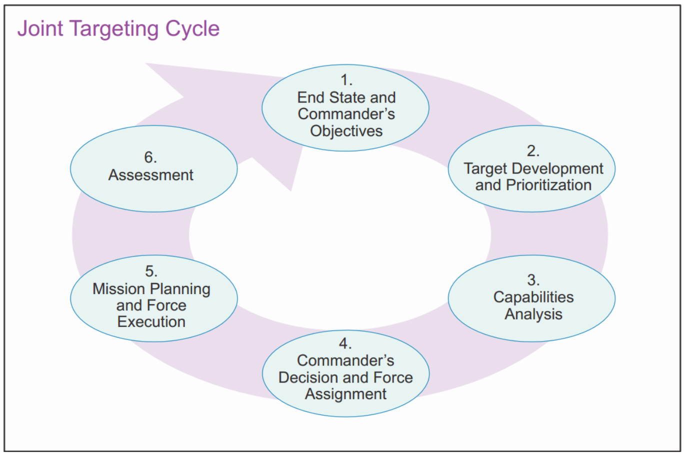

Joint Targeting Cycle
1.
End State and Commander's Objectives
2.
Target Development and Prioritization
3.
Capabilities Analysis
4.
Commander's Decision and Force Assignment
5.
Mission Planning and Force Execution
6.
Assessment

Figure 2.6: Joint Targeting Cycle, from [18, p. II-4]

Targeting focuses on both deliberate targets and dynamic targets. Deliberate targeting focuses on future plans or efforts and is generally 72-hours from execution [18]. Dynamic targeting is conducted against targets of opportunity with emphasis on current operations within a 24-hour time period [18]. The air tasking cycle is a sub-cycle of the mission planning phase in the joint targeting cycle. It converts the targeting plan into an operation order (the air tasking order (ATO)) that assigns attack aircrafts to specific targets followed by a target assessment that confirms the effects achieved [19, p. 81]. It consists of six stages as listed in Figure 2.7.

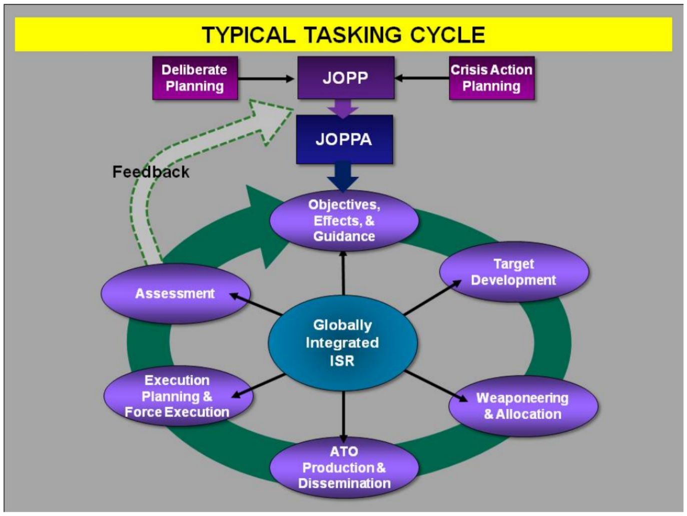
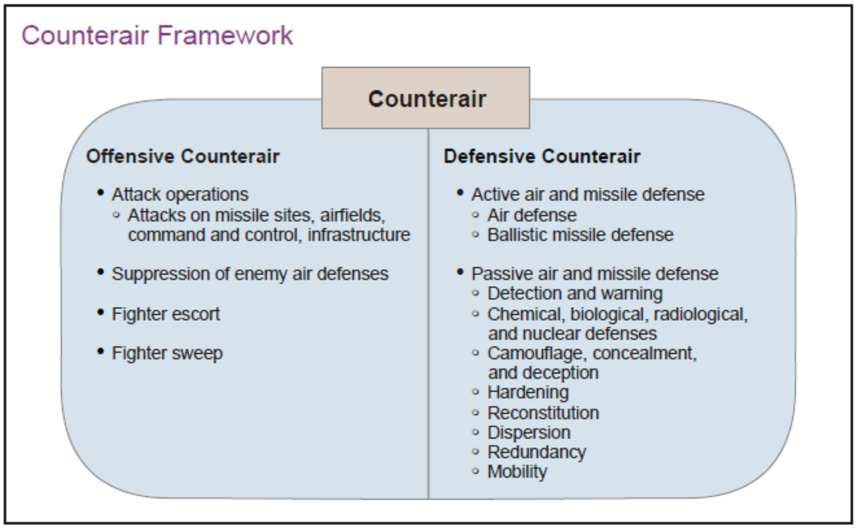


Figure 2.7: Air Tasking Cycle, from [19, p. 82]

### Countering Air and Missile Threats

Air superiority provides U.S. forces with the ability to freely maneuver around the battlefield without being threatened by enemy air assets. *Joint Publication 3-01 Countering Air and Missile Threats* specifically defines air superiority as the "degree of dominance in the air battle by one force that permits the conduct of its operations at a given time and place without prohibitive interference from air and missile threats" [20, p. GL-8]. In order to gain and maintain air superiority, counterair operations must begin as early as possible (e.g., the air war phase of Operation Desert Storm). Counterair is defined as missions that mesh "offensive and defensive operations to attain and maintain a desired degree of air superiority and protection by neutralizing or destroying enemy aircraft and missiles, both before and after launch" [20, p. GL-10]. The counterair framework integrates both offensive counterair (OCA) and defensive counterair (DCA) missions into a single framework. This single counterair framework can be seen in Figure 2.8.



Counterair Framework

Counterair

Offensive Counterair

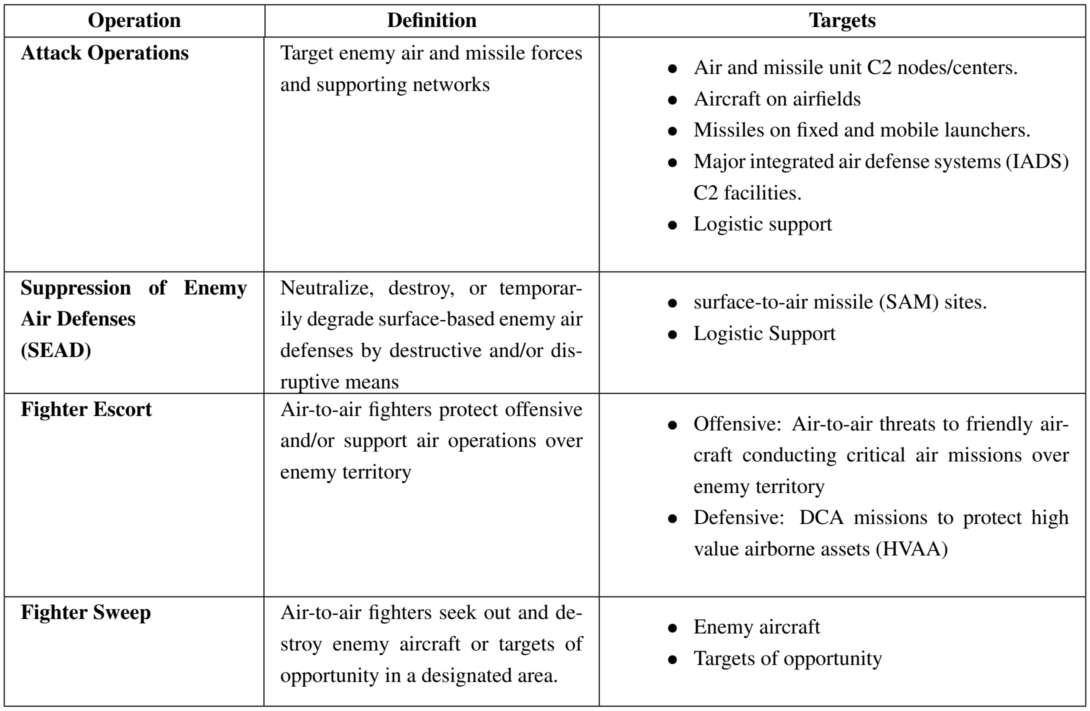
- Attack operations
  - Attacks on missile sites, airfields, command and control, infrastructure
- Suppression of enemy air defenses
- Fighter escort
- Fighter sweep

Defensive Counterair

- Active air and missile defense
  - Air defense
  - Ballistic missile defense
- Passive air and missile defense
  - Detection and warning
  - Chemical, biological, radiological, and nuclear defenses
  - Camouflage, concealment, and deception
  - Hardening
  - Reconstitution
  - Dispersion
  - Redundancy
  - Mobility

Figure 2.8: Counterair Framework, from [20, p. I-3]

OCA operations include "operations to destroy, disrupt, or neutralize enemy aircraft, missiles, launch platforms, and their supporting structures and systems both before and after launch, and as close to their source as possible" [20, p. I-3]. OCA operations reduces the threat to the friendly defenders before those threats affect friendly forces. OCA consists of four operations as listed in Table 2.1.



| Operation | Definition | Targets |
|---|---|---|
| **Attack Operations** | Target enemy air and missile forces and supporting networks | • Air and missile unit C2 nodes/centers.<br>• Aircraft on airfields<br>• Missiles on fixed and mobile launchers.<br>• Major integrated air defense systems (IADS) C2 facilities.<br>• Logistic support |
| **Suppression of Enemy Air Defenses (SEAD)** | Neutralize, destroy, or temporarily degrade surface-based enemy air defenses by destructive and/or disruptive means | • surface-to-air missile (SAM) sites.<br>• Logistic Support |
| **Fighter Escort** | Air-to-air fighters protect offensive and/or support air operations over enemy territory | • Offensive: Air-to-air threats to friendly aircraft conducting critical air missions over enemy territory<br>• Defensive: DCA missions to protect high value airborne assets (HVAA) |
| **Fighter Sweep** | Air-to-air fighters seek out and destroy enemy aircraft or targets of opportunity in a designated area. | • Enemy aircraft<br>• Targets of opportunity |

Table 2.1: Offensive Counterair Operations, from [20, IV-9]

DCA is defined as “all defensive measures designed to neutralize or destroy enemy forces attempting to penetrate or attack through friendly airspace” [20, p. I-5]. DCA provides an area for friendly forces to operate without interdiction by enemy air and missile threats. Examples of air threats include ballistic missiles (BM), bombers, fighter-attack, fighter escorts, intelligence, surveillance, and reconnaissance (ISR), suppression of enemy air defenses (SEAD), cruise missiles (CMs), air-to-surface missiles (ASM) [20, pV-6]. Missile threats entail CMs, BMs which include short-range ballistic missile (SRBM), medium-range ballistic missile (MRBM), intermediate-range ballistic missile (IRBM), and ASM.

### Interdict Enemy Capabilities

The effects of fires is measured in depth of both space (geographical distances and positions) as well as time (current and future). Interdiction, defined by doctrine is “an action to divert, disrupt, delay, or destroy the enemy’s military surface capability before it can be used effectively against friendly forces, or to otherwise achieve objectives” [21, p. I-1]. Interdiction is the action that provides the operational reach to achieve the JFC’s objectives. Interdiction prevents "the adversary from using assets at the time and place of his choosing" [21, p. I-2] through the operations listed in Table 2.2. Each service component possess capabilities that contribute to the interdiction mission.

| Interdiction Operation | Definition |
|---|---|
| Diversion | Divert enemy forces or assets from critically required operational areas. Consumes resources or capabilities critical to enemy operations in a way that is advantageous to friendly operations. |
| Disruption | Interrupt or impede enemy forces, operations, or systems. Upsets the flow of information, operational tempo, effective interaction, or cohesion of the enemy force or their systems. Targets enemy line of communications (LOC), C2 nodes, key commercial infrastructure, and produces a psychological impact on the enemy. |
| Destroy | Damage the structure, function, or condition of a target rendering it ineffective or useless. |

Table 2.2: Interdiction Operations, from [21]

### 2.2.4 Movement and Maneuver

During any ground fight seizing key terrain in order to successfully conduct offensive and defensive operations is crucial. Movement and maneuver as defined by doctrine is comprised of

> the disposition of joint forces to conduct operations by securing positional advantages before or during combat operations and by exploiting tactical success to achieve operational and strategic objectives. This function includes moving or deploying forces into an operational area and maneuvering them to operational depths for offensive and defensive purposes [16, p. III-27]

As per *JP 3-0* movement and maneuver task include:

1. Deploying or moving joint and/or component force formations within the operational area by any means or mode (e.g., air, land, or sea)
2. Maneuvering joint forces to achieve a position of advantage over an enemy.
3. Delay, channel, or stop movement and maneuver by enemy formations. This includes operations that employ obstacles (e.g., countermobility), enforce sanctions and embargoes, and conduct blockades [16, p. III-27].

### 2.2.5 Protection

The protection functions preserve friendly fighting strength through force protection measures. It does this through active/passive defensive measures, fratricide prevention measures, and disease non-battle injuries (DNBI) prevention measures [16, p. III-29].

### 2.2.6 Sustainment

JP 3-0 defines sustainment as "the provision of logistical (planning and executing movement and support of forces) and personnel support that maintains and prolongs operations." [16, p. III-35]. Logistics integrates and synchronizes support efforts (strategic, operational, and tactical ) both within and outside of a theater of operations (known as inter/intra theater).

Key logistic functions include deployment and distribution, supply, maintenance, and food service [22, p. II-2]. While many of these functions are familiar to audiences in a tactical setting, the scope of the functions increases drastically at the operational and strategic levels of war.

## 2.3 Wargaming and Wargaming Design

Dr. Peter Perla, an operational research analyst with over 30 years of experience in wargaming, defines a wargame as, "a warfare model or simulation in which the flow of events shapes, and is shaped by, decisions made by a human player or players during the course of those events" [3, p. 2]. James Dunnigan, a commercial and professional wargame designer and founder of Simulations Publications Incorporated (SPI), defines a wargame as "a combination of game, history, and science. It is a paper time machine. Basically, it is glorified chess" [23, ch. 1] Professor Philip Sabin, a military historian and professor of strategic studies, uses Figure 2.9 to visualize how wargaming straddles the junction between the activities of military affairs, gaming, and simulation.

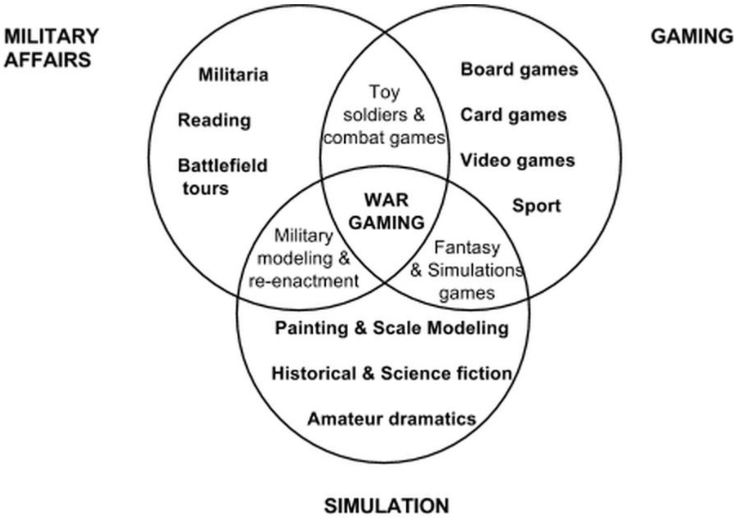

MILITARY AFFAIRS
GAMING
SIMULATION
Militaria
Reading
Battlefield tours
Board games
Card games
Video games
Sport
Toy soldiers & combat games
WAR GAMING
Military modeling & re-enactment
Fantasy & Simulations games
Painting & Scale Modeling
Historical & Science fiction
Amateur dramatics

Figure 2.9: Military Affairs, Gaming, and Simulation, from [24, p. 3]

Sabin states that wargames have two primary components. The first component is a "mathematical model of reality that simulates terrain, deployment/capabilities of military forces, and the passage of time in order to mirror the real range of potential courses and outcomes" [24], while the second component is an "iterative set of active decision inputs by one or more players to guide the simulated actions of the combatants and to respond to the simulated conflict" [24]. Players gain deep insight into armed conflict from these two components. This next section will examine the history, fundamental mechanics, and usage of wargames.

### 2.3.1 Wargaming History

Wargames have a rich a lengthy history dating back to ancient times and have evolved and continue to evolve over the centuries. Some of the earliest examples of board games came from Egypt in the second millenium BC. These were simple stone boards with cells or squares, game playing pieces, and "lots" (early form of dice) to cast. The Romans played a game called *latrunculi* which was similar to chess where players attempted to eliminate their opponents by using two of their pieces to flank the opponent's piece. The Romans frequently carried this game with them as they traveled and archeologists have found copies of latrunculi as far as the British Isles [25, ch 3.1.1]. The Indian game *Chaturanga* is widely considered the predecessor to chess [26]. It uses game pieces divided into four classes, chariots, cavalry, elephants, and infantry, along with two manager classes; the king and the vizier. Each class has its own unique movement system with a goal of capturing the opposing king. *Chaturanga* appeared around 900 BC making its way from the Middle East into Europe and evolved into chess. Chess continued to evolve over the centuries until it became the game know today. Many leaders viewed chess as a game that taught its players military and political strategy by forcing players to anticipate their opponent's moves and plan accordingly. Players began to create a number of chess variations in order to either simplify or add complexity to the game. A German author named Augustus the Younger, drove an effort to refine chess for use as a military simulation which culminated in the game *Kriegsspiel* [25, ch 3.1.1].

In the late 17 to early 1800's the modern wargame-era truly began with the creation of the game *Kriegsspiel* which contained many of the basic principles of wargaming. It evolved the concepts of movement and displacement, unique victory conditions for each opponent, and a color-coded terrain board with combat/movement effects and variable combatants [25, ch 3.1.2]. A modified version of the game was adopted throughout the Prussian military around the 1820s. This version had an open terrain board (no grid squares) and introduced umpires and dice.

The Prussian military as well as the Italians, British, Russians, Japanese, and even Americans continued to use *Kriegsspiel* as part of their training throughout the rest of the 1800's and early 1900's [27, p. 5]. The game continued to evolve with players demanding more realism while designers struggled to keep the game playable. In 1913 HG Wells published a book called *Little Wars* that introduced an engaging playing system with miniature soldiers and spring-loaded cannons. He included an appendix, "pointing out how a Kriegspiel of real educational value for junior officers may be developed out of the amusing methods of Little War" [28]. This balance between realism and playability remains a major balancing issue for modern wargame designers.

Wargames continued to evolve, driven by military training demands. After World War I (WWI), militaries around the world used them to test battle plans and gain insight into their enemy's thinking. The Germans, in particular, used wargaming before every major operation of World War II (WWII) [27, p. 7]. Prior to WWII, there was only a limited civilian market for wargaming, primarily due to the cost and complexity of the games.

There was no real delineation between the civilian and military games. This symbiosis changed following WWII due in part to the introduction of the atom bomb which many civilians found difficult to accept and even more difficult to role-play in a game. In addition, the military began to transition from paper wargames to more expensive computer assisted wargames causing the two markets to diverge. The civilian market preferred historical combat while the military market focused on present-day and future conflict scenarios [25, ch 3.1.7].

Beginning in the mid-50s a number of civilian wargame publishers appeared to support the separation of the military and civilian wargame markets. One of the earliest commercial companies is Avalon Hill, who published their first wargame, *Tactics*, in 1954. *Tactics* brought the combat resolution table (CRT) to the civilian market as well as a much faster pace than military specific wargames. Avalon Hill produced a number of additional games such as *Gettysburg*, *U-Boat*, *Chancellorsville*, etc. From the 1960s to the 1980s several other companies such as SPI, Victory Games, and Game Designer's Workshop (GDW) appeared and released a flood of commercial wargames that covered a vast array of past and present war battles or campaigns representing all three levels of war. They gained considerable popularity although it remained a niche hobby even at its peak.

Commercial wargames often combined a paper or mounted map, game counters on cardboard, a CRT, and dice to act as a random number generator. These games possessed varying degrees of complexity which, as time passed, became increasingly more complicated and less approachable for new players. The time required to play a wargame went from hours to days or even months. Players struggled finding opponents due in part to the difficulty in scheduling gameplay sessions. By the late 1980s and 1990's the civilian paper wargame market took a downturn, in large part due to the introduction of the video game and computer game market and partially due to the industry's self-inflicted overemphasis of marketing to the hardcore wargamers rather than the casual players [23].

Ironically, computers, which were one of the sources of the civilian wargame market's downturn in the 80s and 90s, are helping the industry experience a resurgence. The connectivity of the Internet provides game designers a direct connection with players and makes it easier for players to locate, pre-order, and purchase wargames. Time and time again wargames evolved to adapt to the needs of the players. This evolution occured through the hands of game designers that recognized the needs of the players and experimented with game design to meet those needs.

### 2.3.2 Wargaming Design Fundamentals

Phillip Sabin a British military historian and professor states,

> The most important function of wargames is to convey a vicarious understanding of some of the strategic and tactical dynamics associated with real military operations [24, p.31].

Wargames teach through active learning as wargamers ponder strategic, operational, and tactical problems to overcome their opponents. Players also learn that the historical outcome of a battle was not predestined, but rather dependent on a host of variable factors [24, p.36-37].

Peter Perla identifies wargames consisting of objectives, a scenario, a data base, models, rules, players, and a post-play analysis [4, ch. 4]. These elements are summarized in Table 2.3.

| **Wargame Element** | **Definition** |
|---|---|
| Objective | Set of goals for the players to achieve from playing the wargame |
| Scenario | Specific situations and context for decision making |
| Database | Geographical information: Map<br>Order of Battle: Units within the scenario<br>Situational Information: force arrayment, weather data, victory conditions, etc. |
| Models | A combination of look-up tables and mathematical expressions that translate the game's data and the players' decisions into game events |
| Rules | Procedures that dictate how and when to apply the models |
| Players | Humans whose decisions affect and are affected by the flow of game events |
| Analysis | Assures the capture of the game's objectives and gains understanding of why decisions were made |

Table 2.3: Wargame Elements, from [4, ch. 4]

Wargames integrate numerous models of conflict simulation and must judiciously walk the line between accuracy and simplicity. This becomes more difficult as additional models are added to the wargame [24, ch. 2]. For example, balancing accuracy and simplicity is easier for a wargame that focuses exclusively on ground warfare versus a wargame that focuses on both ground and air warfare. This balance is represented throughout the entire game design, particularly in the level of resolution.

One example of a resolution decision is the map design and scope or the general level of artistic representation of the components. Another example of accuracy versus simplicity is the number of counters. Typically a 2-hour game should have no more than a combined 40-50 counters for both players [24, p.79]. There are countless examples of wargames that have 100s of counters and require months to play through a single campaign demonstrating the very inaccessibility that plagues both the commercial and military markets.

## 2.4 Current Digital Wargames

Numerous digital and paper wargames exist that offer inspiration and insight into good game design. This section briefly describes a few games analyzed during the research phase of this thesis. User interfaces, complexity, and web-based gaming capabilities are viewed through the lens of each game discussed.

### 2.4.1 Battle of the Bulge

Wargame designers often attempt to harness modern computers' processing power to increase the resolution of their games by adding additional historical details. This additional detail can adversely affect these games by increasing complexity and reducing playability. A better transition for physical wargames to the digital domain may be touch-based tablets.

*Battle of the Bulge* (BotB) is an operational level wargame on iOS set during the titled campaign [29]. Originally designed for the iPhone, BotB is a traditional wargame that portrays MCO and takes full advantage of the touch device interface [30]. Wargames are highly detailed, and the amount of information a player is required to understand and process can be daunting. BotB uses a tablet's entire screen to display the map board and all playable units. Game menus and notification windows enter the player's view from all sides of the screen and stack on top of the map based on priority. This creates an intuitive interface immersing players in simplified gameplay.

BotB's casual appeal is generated by limiting the number of actions a player can conduct.

Unit movements are restricted to one, two, or three large area movements on a map divided into regions. These large area movements simplify the decisions a player has to make and allows the units to reach their goal at a reasonably fast pace. Players are only allowed to move units once within a 24 hour (game time) period, reducing the number of actions a player can take each turn. Although complexity is minimized, strategic understanding is not. Players must think multiple moves ahead since an activated unit, one that moves or initiates an attack, will not be available for an extended period of time. Using area movement and reducing the number of actions a player can take each turn makes BotB an engaging casual touch-based wargame.

### 2.4.2 Panzer Campaigns

If BotB is a lesson in simplicity, John Tiller's *Panzer Campaigns* series is a lesson in complexity [31]. Games in the series were initially developed for the personal computer (PC) but have been ported to tablet devices in recent years. Traditional paper based wargames, as discussed earlier, try to replicate many components of battle from maneuver, logistics, fires, etc. Using the power of computers, the *Panzer Campaigns* series is a true attempt to recreate detailed paper wargames on digital devices.

The *Panzer Campaigns* series games are turn-based and set during WWII. Unit counters typically represent Army companies or battalions, placing the games at the tactical level of war. A player's individual turn can take up to two hours to complete [31]. The games use hex-based maps populated with prepositioned units ready to do battle. A menu at the top of the game screen provides numerous graphically labeled buttons that can be difficult to interpret for beginners. Plenty of reading material is provided in the form of rule books and user manuals. After reading through the manuals and deciphering the controls, it is possible for a committed player to learn how to play the game. The target audience for John Tiller's *Panzer Campaign* series are the more experienced wargame players. Consequently, novice players may have a difficult time adjusting to the high level of complexity of both the user interface and the game design.

### 2.4.3 OpenPanzer

*OpenPanzer* is is a remake of the old PC game *Panzer Generals 2* developed in the late 1990s [32]. *OpenPanzer* is open source and free to play online. Since it is a direct replication of the original, *OpenPanzer* is a notable example of a professional looking web-based wargame. It displays how much can be accomplished in creating a robust two-dimensional wargame in the browser.

*OpenPanzer* makes itself accessible to players both new and old by providing useful tutorials and intuitive controls. The game is hex-based which is not obvious when looking at the map. When a unit is selected, hexes come into view displaying available movement options. Techniques like this significantly reduce the complexity of learning the game. Reducing difficulty in learning how to play a wargame must be accomplished if it is intended for casual players and not just the traditional hobbyist.

## 2.5 Coding the Web

Computer programming languages are becoming easier to learn everyday. WARLOC was developed using HTML5, JavaScript and some Cascading Style Sheets (CSS). These different coding languages are briefly introduced as well as some existing JavaScript code libraries. Sources cited can provide the reader with more information though a simple Google search of any of these terms will return more sources than can be explored in a single sitting.

### 2.5.1 HTML5, JavaScript, CSS

When developing web sites or web applications, familiarity with HTML5, JavaScript and CSS is needed. HTML5 is the latest version of HTML which is ultimately responsible for the structure of all web pages found on the Internet. CSS is a language used to stylize modern day web pages. Finally, JavaScript, is an asynchronous functional language that provides much functionality to today's dynamic web pages [33].

Learning how to use any of these languages is incredibly easy with today's Internet. A simple Google search of any of these terms will lead to numerous books, papers, blogs, etc. Web sites like, khanacademy.com [8], codecademy.com [34], or w3schools.com [35] offer free classes in HTML, JavaScript, etc. Once familiar with a language, numerous communities exist that assist beginning programmers any time they get stuck. Sites like stackoverflow.com [36] have coders who ask questions in a blog type setting that are, often quickly, answered by other coders. In addition to various learning sites on the web, future coders need not start from scratch as various JavaScript libraries exist waiting to be utilized.

### 2.5.2 JavaScript Libraries

Atom.js, released by Chris Campbell [37], is a useful game development library. Atom.js provides a basic game function that represents the general architecture of a typical game. It consists of various subfunctions to include update, draw, run, stop, and step. Each of these functions are named based on their purpose. For example, the draw function draws graphics to the screen while the step function allows the programmer to control the program's flow through different functions of the game as needed. The overall game function is repeatedly looped through executing each of these functions until the game reaches its end state.

EaselJS is a JavaScript library that hooks into a developer's HTML page through its ”<canvas>” element. The library can be used to draw directly on the canvas or through the use of a ”Stage” function. Programmers can add functions, arrays, or JavaScript objects to the stage using the ”addChild()” method. The library is well documented and has an organized website for its application programming interface (API) [38]. The library also contains functions to handle mouse clicks, mouse hovers, and mouse drags. Any future web-based wargame developer would be interested to know that EaselJS does contain a simple function for creating hex shapes.

GameJs [39] is a small JavaScript library with dual functionality. It simplifies canvas manipulation and provides basic gaming mechanics. Methods to manipulate the canvas include the ability to draw shapes, create sprites, and create multiple surfaces or layers. The library provides easy handling of keystrokes and a preload function for game resources such as image files.

Another useful and well documented game library is melonJS [40]. MelonJS works well with programs like Tiled which is used to develop maps and sprites. The combination of melonJS and Tiled leads to easy development of side scroll games but could also be useful in developing web-based wargames.

The libraries discussed above are only a snapshot of what is available. All the libraries are under open source licenses. This makes them available to anyone who wants to utilize them within the terms of their respective license. In addition to being familiar with a library's license, developers should understand that there is a significant learning curve when using any of these libraries. Some libraries require the use of a very specific syntax which may or may not be easy to learn. If time is not a factor and a library offers useful complex functions then it is definitely worth the investment. However, writing one's program using core JavaScript is completely unrestricted by any license and will often result in a brand new library of functions that, if coded correctly, can lead to code reuse on future projects.

## 2.6 Open Source Tools

In the early 1970s, developers attempted to convert paper wargames into computer wargames but the hardware was expensive and the development tools were primitive [23]. Today, hardware is faster and cheaper. Numerous advanced development tools are freely available on the internet. Some of these tools include code editors, collaboration tools, code repositories, version control systems, photo and graphic editing software, and game engines.

The cost of Open Source Systems (OSS) is very attractive when analyzing the Total Ownership Cost (TOC) of both systems and software. DoD policy does not prohibit the use of OSS as clarified by the DoD Chief Information Officer (CIO) in 2009 [41]. In fact, research found that, in regards to the U.S. Air Force's Standard Desktop Configuration, "OSS offerings provided what might be called the 90% solution" [42, p.60]. Regardless, it is not easy to install software on a government system. Certain steps must be taken to ensure that the software meets all Information Assurance (IA) requirements and earns a certificate of networthiness (CoN). On the other hand, some tools discussed in this section exist online and and require no local installation. If a government computer is allowed unrestricted access to the site by the local Network Enterprise Center (NEC), then the application can be used on any government computer. This is important since acquiring a CoN for a piece of software is yet another requirement placed on Army units that can be labor intensive, require many months to complete, and take time away from units' training schedules.

### 2.6.1 Brackets

Brackets is an open source web development code editor for HTML, CSS and JavaScript [43]. Brackets is licensed under the MIT License [44] providing programmers the freedom to download and use the software for any legal purpose they desire. Since the source code is maintained by Adobe and is openly available for review, developers should be comfortable downloading and utilizing this editor.

The Brackets editor recognizes and color codes keywords of JavaScript, HTML, and CSS code. Brackets offers code completion initially starting with basic core JavaScript functions and building its library as the developer writes their code. Brackets also provides a live preview window that allows a developer to watch their edits happen in real time on the webpage as they are typing the code. This not only reduces development time but also makes it easier for novice coders to quickly become more familiar with the code they are using. Brackets is supported by the open source community resulting in bi-weekly updates to the editor.

### 2.6.2 Cloud9

Cloud9 is a cloud based integrated development environment (IDE) [45] containing many of the same features as Brackets but is focused on cloud-based collaboration. Since Cloud9 is cloud-based, no software installation is required. By placing developing code on the cloud, multiple programmers can actively edit code in real time. This removes the old concept of checking out a file from a central repository, making edits, and reposting the changes. Instead, all developers on a project are simultaneously present in the coding environment all at once and can see each other's cursor location. Cloud9 also provides a chat feature allowing discussion within the coding environment.

### 2.6.3 GitHub

Since software projects can be large and require multiple people to work on multiple parts at once, some form of control is required. A version control system places the code in a repository and tracks all changes to the code. The system further allows teams to roll back to a previous version if it is determined that the path taken by the development team results in a dead end. One such version control system is known as Git [46] which is the foundation of GitHub, "a web-based hosting service for software development projects that use the Git revision control system" [47].

GitHub has been labeled as "a place for social coding" [48]. Many of the code repositories listed on GitHub are freely available as it offers free accounts and storage space to those who agree to make their code open source. Many educational books that teach code development use GitHub to post the source code used within their published text. On the private side, GitHub offers repository hosting services for a fee based on the amount of space required. Overall, GitHub is an inexpensive version control system worth considering for any future software projects.

### 2.6.4 Google Docs and Google Drive

Google Docs and Google Drive are two up and coming cloud-based collaborative tools. Both are actively supported by Google's staff and are accessible wherever Google is accepted. Google Drive provides cloud-based storage of personal files to include Microsoft Word, Excel, PowerPoint, etc. Google Drive allows users to store files on the cloud and can grant access to those who need it. Google Drive even offers version control though not as robust as something found in GitHub or even Microsoft SharePoint. While the ability to store and share files for free is impressive in itself, the true innovation lies in Google Docs.

Google Docs is a productivity suite similar to Microsoft Office. Documents, spreadsheets, and slide presentations can all be created as a Google Doc. The added benefit here is, once again, online collaboration. If a user posts a Microsoft file to Google Drive and shares it, only one person can edit the file at a time. If the person elects to convert the file to Google's format, the file can be edited by multiple people at the same time. This is similar to the collaboration capabilities Cloud9 provides with code but with traditional documents.

In addition to traditional Microsoft Office style documents, Google Docs also offers a graphics editor. Though not as robust as Adobe Photoshop or the GNU Image Manipulation Program (GIMP), the Google graphics editor offers an impressive amount of features. Features that allow the creation, for example, of unit counters for a web-based wargame.

### 2.6.5 GIMP

GIMP is open source software that allows photo editing and graphic creation [49]. Under the GNU license, GIMP is free to use and free to modify [50]. As an open source program, GIMP's interface is not as intuitive as an Adobe or Microsoft interface and can be difficult to initially learn and use. However, the learning curve for GIMP is easier to digest than the cost of a single license for an Adobe or Microsoft product.

### 2.6.6 VASSAL

VASSAL is a Java developed open source game engine used to adapt paper wargames into a digital format [51]. This translation is quite literal as VASSAL provides no computer logic. A digital wargame with programmed logic would be expected to automate the task of rolling dice and automatically use its CRT to resolve combat. VASSAL does not do this. Instead, VASSAL provides a random number generator that has to be manually clicked by the player who then adjudicates combat by manually looking up the results of the generated value in a digital CRT. If a unit counter takes a step loss or is destroyed, the players manually flip the counter or drag the counter to the graveyard, using the mouse.

Each digital board game created in VASSAL is called a VASSAL module. VASSAL module sizes are typically governed by the size of their associated images (game map, CRTs, unit counters, etc.). The modules can be loaded to the VASSAL servers for distribution or can be sent between players using email or some form of physical media such as a CD-ROM or a USB drive. VASSAL provides a play by email (PBEM) function that allows dislocated players to play over the Internet [52]. The main drawback to VASSAL is that anyone wanting to play a wargame module has to download and install the VASSAL engine. This poses issues for the military as it requires a CoN in order to install VASSAL on a government computer.

Once a wargame is developed into a VASSAL module it can act as a very base prototype for the development of an automated version of the same wargame. Developers have a basic look and feel for how the game should operate. By playing the VASSAL module it becomes clear what functions will need to be automated such as flipping and destroying counters. Since VASSAL modules contain the all the game components (counter images, game map, rule book, etc.) it can act as the GDD for development of an automated version of the wargame.

## 2.7 Browser Power

Compiled languages and interpreted languages are two prominent programming languages in existence today. Compiled languages take a programmer's written code and translate, or compile, it into an executable file understood by the computer's hardware. In relation to web applications, this executable file must be downloaded to the host system, often in the form of a plugin, in order for the host system to execute the program. This has been labeled by some as the "compiled web model” [53].

Interpreted languages on the other hand, are executed through the use of a virtual machine or interpreter. These virtual machines read a programmer’s code at run time and translate it into machine level instructions which are then executed by the local machine. Having this extra step of translation at runtime has a performance cost but is continually being reduced by constant technological advancements. These advancement have led to a concept labeled by some as the "interpreted web model" [53].

The big advantage of the interpreted model is the idea of executing code in the browser without having to download an executable. Most modern web browsers contain an embedded JavaScript interpreter that allows JavaScript code to be translated and executed by the browser itself. Application updates can be done in a single location. There is no need to deploy the update as each client will receive the modified code the next time they connect to the server in which the code resides. Though this makes it possible for numerous web applications written in JavaScript to execute on almost any modern computer, internet browser compatibility is still a challenge. Browsers such as IE, Google Chrome, etc., all have unique functionalities and can interpret a programmer’s code differently.

Downloading and running executable files from the internet is discouraged, if not blocked, on government information systems. Client-side JavaScript contains self-imposed limitations and functionality. For example, Javascript does not have the native functionality to read or write files to the client computer. A quick summary of JavaScript security is provided in O’Reilly Media’s JavaScript: The Definitive Guide, 5th Ed [7].

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 3: WARLOC Design

The overarching goal of this thesis was to create a web-based wargame rooted in military doctrine to familiarize young officers with what many believe is becoming a lost art, the execution of MCO. Prior to writing a single line of code, a paper version of the game was designed. Designing a wargame is an extremely complex and challenging endeavor. It requires a clear vision to translate feasible learning objectives into a game that is both functional and worthwhile. The designer must balance accuracy and simplicity throughout all elements of the wargame design in order to achieve these objectives.

Understanding wargame design is a valuable discussion to have within the military. We contribute to this conversation by examining design elements of our wargame titled Web-based Army Repeatable Lesson in Operational Combat (WARLOC).The design discussed this chapter led to the creation of a paper game, a VASSAL module, and finally the web-based version presented in this thesis.

## 3.1 Wargame Format

A JFC leverages a number of methods to train personnel on military operations and implementation of the joint functions. The methods used include staff rides, field training exercises, command post exercises, and professional development sessions. One aspect that many of these methods share is their use of professional wargames to enhance training. These professional wargames are typically very complex. This complexity resides within highly detailed models, in-depth scenarios, complicated rule sets, and elaborate databases. These games often have a large support requirement in terms of manpower, time, and cost. While the professional wargame offers tremendous detail, it is neither quick nor easy to implement and utilize. This barrier prevents the simple plug-and-play learning experience that a time-constrained junior officer requires.

A casual wargame is an excellent way to familiarize officers on the basics of MCO at the operational level as they can lead to a higher acceptance rate. ”Casual games can only be loosely defined as those titles that are friendly to new/occasional users and are intuitive and accessible" [54]. Intuitive interface, accessibility, simple mechanics, and short gameplay length are all characteristics of a casual game. These characteristics support a training audience entering an extremely busy stage in their careers with limited free time. In consideration of this limited free time, WARLOC was designed to take no more than two hours to set up and play.

The passage of time had to be represented in WARLOC. Two basic methods of time representation in games are real-time and turn-based [55]. In real-time strategy games, the game time progresses continuously without pause, generally at the same pace as the outside "real-world." Game events, such as enemy attacks, occur whether players make a decision or not. Conversely, in turn-based strategy games, time is divided in regularly occurring increments and only progresses once the previous player completes their turn.

There are advantages and disadvantages to both systems. Real-time strategy is very common in modern games and is an excellent method to force players to make rapid decisions. Turn-based strategy games allow players to progress through the game at their own pace and along their own lines of prioritization [55]. WARLOC is targeted at an audience with little to no experience with MCO at the operational-level. We want the players to think through the complexities of planning and executing each of their decisions. Thoroughness is more important than speed. Therefore, we chose to use a turn-based system for WARLOC.

No fog of war exists in the paper version allowing players to observe each others moves. This provides each player additional time to develop a strategy in response to their opponents moves. Constant focus on the game, the situations presented, and strategic thought process truly immerse the player in the game. If the opponents are well matched, flow may be introduced.

Flow is defined as "the state in which people are so involved in an activity that nothing else seems to matter, the experience itself is so enjoyable that people will do it even at great cost, for the sheer sake of doing it" [56]. Jane McGonigal discusses the power of games in terms of flow by saying,

> If the goal is truly compelling, and if the feedback is motivating enough, we

> will keep wrestling with the game's limitations —creatively, sincerely, and enthusiastically— for a very long time. We will play until we utterly exhaust our own abilities, or until we exhaust the challenge. And we will take the game seriously because there is nothing trivial about playing a good game. The game matters [57, p. 27].

We wanted to harness the power of flow through WARLOC. The goal of WARLOC's design is to have two near peer opponents, equal in skill, to engage in operational warfare. We wanted victory for each side to be challenging but not impossible. If designed properly, WARLOC will cause players to intensly focus on the game itself. If the players are equally matched, the two hour play time will pass unnoticed. If the players have a strong competitive nature, repeated play could occur. This can result in players immersing themselves in the subject matter for hours on end. Harnessing the power of flow in a fun gaming environment while presenting doctrinal concepts avoids boredom leading to more effective learning [58].

Paper and computerized versions of casual wargames are the two most prominent formats of wargames today. Each has its merits as well as its limitations. However, when reviewed as a whole, the computerized versions offer many advantages for a casual wargame. One of the key requirements is accessibility. Players can access a web-based game through any browser on any computer with access to the Internet at no additional cost. In this respect, the advantages of a web-based version significantly outweighs the traditional board game version. Another advantage to a web-based game is the reduced setup time. A computer wargame can use a predetermined scenario that includes prepositioned forces immediately at load time. Typically, in a board game version, game setup consumes anywhere between 15 minutes to over a hour. Some paper wargames known as "monster" wargames can contain numerous maps and a very high number of counters making setup times easily exceed a single hour. Board games often provide players with "aid" cards, or cheat sheets, but often times these are overly complicated and difficult to interpret. A properly designed computer wargame possesses an interface that can anticipate the needs of the players and presents relevant information which in turn facilitates a player's decision-making process. These three characteristics demonstrate the advantages of using a web-based computer wargame.

## 3.2 Wargame Elements

As discussed in subsection 2.3.2, Peter Perla uses a wargame design framework consisting of seven elements [4]. The following sections will use Perla's framework to explain WARLOC's concept and design.

### 3.2.1 Objectives

The learning objectives drive the focus of all wargames. Clear, understandable, and well-defined objectives are first determined in the design process [24, p. 37]. This is equally true for educational, training, and entertainment wargames. The overall goal of WARLOC is to assist junior officers to prepare for service in a joint operational command. This is accomplished by providing a basic and casual web-based wargame focused on the operational level of combat. The following learning objectives drive the overall game design of WARLOC.

#### Familiarization with Major Combat Operations (MCO) at the Operational Level

The overall objective of WARLOC is to introduce MCO to junior officers who have served in a military entrenched in a COIN environment. Junior officers today have become very proficient in tactical thinking, the foundational level of warfare. In order to serve in future positions as field grade officers, junior officers must increase their understanding of both the operational and strategic levels of warfare. WARLOC combines MCO with operational-level decisions in an attempt to familiarize junior officers with the operational level of war and reinforce this familiarization through repeated play. All remaining learning objectives support this overall learning objective in some form or fashion.

#### Familiarization with Joint and Component-level Task Organization

Junior officers serving at the tactical level often have little to no experience with any military service branch other than their own. They do not understand the role or the organizational structure of other service branches. In WARLOC, players will command a combined joint task force (CJTF) consisting of both ground and air components from two nations along with their respective subordinate commands. By directing ground corps and air wings, players are introduced to the make-up and role for the Army and Air Force service branches from four different nations.

**General Understanding of the Joint Functions of Fires, Movement/ Maneuver, and Sustainment**

Players are typically indoctrinated in their own service's joint functions-equivalent. For example, the Army's concept of the joint functions are known as warfighting functions. These warfighting functions, while similar to the joint functions, are specific to the Army's mission and capabilities. Players may be unfamiliar with the additional capabilities and requirements that other services provide in support of the joint functions. WARLOC will give players hands-on experience with the fires, maneuver, and sustainment joint functions and exercise their ability to leverage them in a MCO environment through active planning and execution within the game.

**Introductory Experience with Air Power in a Joint Task Force**

Army officers typically have very limited experience with airpower. Any experience is almost exclusively with close air support (CAS) and neglects many of the capabilities and tenets of airpower which include flexibility, persistence, and concentration [59, p. 37]. WARLOC requires players to intelligently plan and execute high-level air operations if they hope to achieve victory. In turn, the player receives a basic understanding of the implementation and capabilities of airpower in a combat operation.

### 3.2.2 Scenario

A game's scenario is crucial to providing the setting and context of the game. It gives the game epic scale, making the player feel like the decisions they make within the game have more meaning [57]. WARLOC's scenario places the players in a position to make decisions often made by general officers. There are currently a limited number of MCO threats to the United States. However, recent events demonstrate the potential for a North Korean threat [60]. WARLOC's scenario is set in a hypothetical future, the year 2022, with a revitalized North Korean economy. North Korea's economy has led to a major upgrade in its military force allowing it to conduct an invasion into South Korea. WARLOC's scenario reads as,

> After months of provocations from both sides of the Military Demarcation Line, the United Nations Security Council is at a deadlock. Both China and Russia are staunchly against any United Nations (UN) or Republic of Korea

> - United States (ROK-US) offensive operations into North Korea. Seizing this period of indecision, North Korea launches an attack into South Korea with the intent to conduct a swift war while threatening nuclear retaliation if ROK-US forces enter North Korea. The People's Republic of China (PRC), while against a North Korean invasion into South Korea, is also intent on preventing a nuclear engagement. In response, the PRC is supporting the defense of North Korea by augmenting North Korea's defensive ground and air forces. In addition, they are providing a naval presence in the Korea Bay off of North Korea's western coast. Meanwhile, Russia conducts naval "exercises" in the Sea of Japan directly off the eastern coast of North Korea effectively preventing ROK-US naval operations. Appendix B.

Players play the role of either the Combined Forces Command, Korea (CFC), the ROK-US alliance that defends Seoul, or as the fictional Juche Alliance of North Korea and the PRC attempting to seize Seoul. Since WARLOC's focus is on familiarization of MCO rather than determining the best counter-strategy for a North Korean invasion, the scenario was tailored to create a true peer competitor to the U.S. Since this is a "what if" scenario, those who do not believe in North Korea's ability to invade South Korea can rest easy and enjoy the game as a work of fiction. Those who do believe this scenario is on the horizon will appreciate the fact that it is set in the non-too distant future. It should be stated that wargames are not intended to predict the outcome of battles in any shape or form. WARLOC is no different. Wargames simply provide insight, expose decision makers to tough scenarios and allow them to think through the problem in a safe setting. When presented with a different but similar dilemma in the real world, decision makers will be more mentally prepared. This is the power of wargames, nay, this is the power of all games. They force players to think critically.

### 3.2.3 Database

Attempting to influence gamers outside of the game can risk a player's satisfaction with the game itself [61]. To minimize this effect while still trying to exploit a players motivation to play the game, a database of necessary and relevant data is built as the foundation of the wargame. The accuracy of a wargame's database leads to the accuracy of its models. Accurate models produce authentic in-game situations forcing players to make decisions similar to those made during actual war. This engagement with the game can produce insightful and interesting dilemmas. The more interesting or radical the in-game situation, the greater the association or link with the decision made, which in turn can lead to greater retention [62].

WARLOC's database consists of task organization, combat and logistic capabilities of assigned forces, and geography. This database is used to determine the combat power of individual units, movement capabilities due to terrain and supply lines, as well as effects of critically damaged support infrastructure such as logistics hubs.

### Task Organization

Players control either the CFC or the Juche Alliance. Both are combined JTF consisting of joint military forces from two different nations. Figure 3.1 and Figure 3.2 from Appendix A show the top level task organization of each of these JTFs.

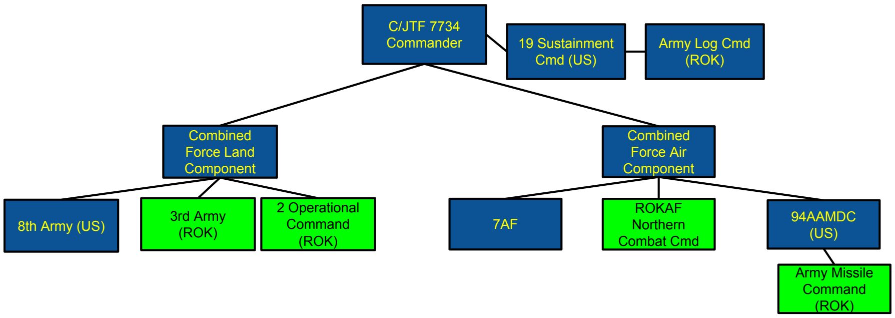

C/JTF 7734 Commander
19 Sustainment Cmd (US)
Army Log Cmd (ROK)
Combined Force Land Component
Combined Force Air Component
8th Army (US)
3rd Army (ROK)
2 Operational Command (ROK)
7AF
ROKAF Northern Combat Cmd
94AAMDC (US)
Army Missile Command (ROK)

Figure 3.1: Combined Forces Command Task Organization

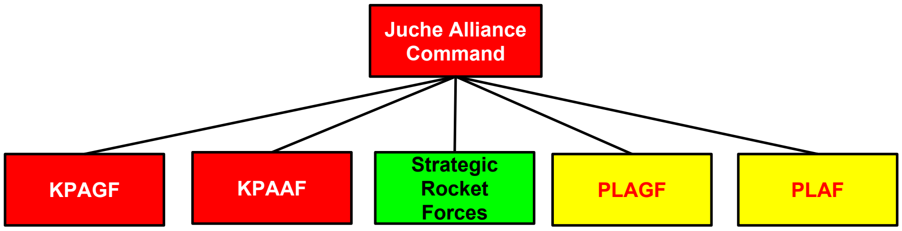


Figure 3.2: Juche Alliance Task Organization

The CFC forces are divided into a Combined Forces Land Component Command (CFLCC) and Combined Force Air Component Command (CFACC), along with a separate sustainment command. In contrast, the Juche Alliance has less experience with joint operations and keeps its Korean People's Army (KPA) and People's Liberation Army (PLA) ground and air forces keenly divided by national service branch rather than as combined/joint components.

Both the CFLCC and CFACC contain major subordinate commands (MSCs) that are typically either two-star or three-star general officer billets and link the operational and tactical levels of command. Within each of these these MSCs are Republic of Korea (ROK) and U.S. O-6 brigades or air wings. (See Appendix A for the MSC task organization charts).

The Juche Alliance's task organization is quite different in several ways. For example, the lowest echelon for combined operations is at the JTF-level of the Juche Alliance Command headquarters. In reality, even this level of combined operations might not occur depending on the diplomatic agreement between China and North Korea. The Juche Alliance component-level forces are separated along national service branch lines. The two air component commands are the Korean People's Army Air Forces (KPAFs) and the People's Liberation Army Air Force (PLAAF). These two commands would most likely have clearly delineated areas of operation and experience confusion when events occurred along the seams and gaps of these areas. The same relationship occurs between the Korean People's Ground Force (KPAGF) and People's Liberation Army Ground Force (PLAGF). Lastly, the strategic rocket forces (SRF) are responsible for the theater ballistic missile (TBM) fight.

### Combat Power and Logistic Capabilities

Creating a comparative measurement of combat power and logistic capabilities between two forces is a critical step in game development. It defines the balance between the two opponents and drives a great deal of the strategy that the two players apply. WARLOC represents both air and ground combat which requires both a comparison between similar forces and between opposing air and ground forces. The benefit of this comparison is that it provides the players with a perspective of the strengths and limitations of the components in a joint organization.

WARLOC uses unit counters with a series of values that represent the comparative measurements of combat and logistic capabilities. The combat capabilities are represented by values for combat power for direct fire systems, indirect fire, and a series of aircraft based capabilities (air-to-air, SEAD, air-to-ground). Logistic capabilities are measured in a unit's movement allowance, which measures the amount of movement a unit may conduct in a given turn. We assembled an initial set of values based off of unclassified open-source research of unit types and capabilities and then, through a series of playtesting and refinement, re-scaled the numbers to improve the balance of the game. (Further explanation of this system is in Appendix B.

### Geography

Two primary factors considered when designing the map were how far beyond the battle area to draw the map and how to divide the map terrain. The game's scenario takes place on the Korean peninsula and drives most of the interaction of the opposing forces between the demilitarized zone (DMZ) and Seoul. The game map presents the area of the Korean peninsula between Pyongyang in North Korea and Daejeon in South Korea. This gives sufficient space for staging military forces, urban terrain, air bases, seaports, and road networks. It also includes a small coastal region on both the western and eastern perimeters, although there are no naval forces in the game.

A designer may divide a map's terrain into zones or not divide a map terrain at all. A zone-free map allows players to move anywhere they desire within a given radius. However, the number of possible moves is nearly endless which slows down gameplay as players internally debate their next move. Using zones offers the advantages of reducing the number of possible moves that a player must choose from. Options for zones include regular zones (squares or hexes), irregular zones (varying large and small areas), or a hybrid (i.e. hexes mixed with large/small irregularly-shaped areas) [24, p. 69-75]. A hex-based map was chosen for WARLOC matching nicely with the shape of the Korean peninsula. The map's layout was modeled after the map used in the GMT game, *Next War: Korea*. The landscape is divided into four terrain categories that have differing effects on maneuverability for ground units (refer to Appendix B for further details). The added benefit of a hex-based map is that it makes it easier to measure unit counter movement and weapon range distances.

Key terrain is an important consideration in combat. Securing a flat piece of land to construct a make-shift airfield to allow for the arrival of reinforcements and supplies can be critical to mission success. By varying the terrain types on the map, players are provided with another level of strategy to consider. In reality, forward edge battlefield areas (FEBAs) provide defensive barriers making it difficult for Juche Alliance forces to maneuver and are an important in the defense of Seoul. This is simulated in the game as attacking Juche Alliance ground forces are forced to cease movement once entering a FEBA hex, no matter how many movement points still remain. In addition, invading forces from the North are not able to maintain reinforcing positions and are thus at a disadvantage when attacking from a FEBA. This is represented in the game as Juche Alliance units attacking out of the FEBAs receive a -2 dice roll modifier (DRM) which reduces the value of a rolled dice by two. This makes it very advantageous for U.S. forces to lock Juche Alliance forces in FEBA hexes utilizing their zone of control (ZOC).

These terrain effects help in the pacing of the game. FEBAs prevent non ROK-US forces from rushing to Seoul within two turns. The mountains help to channel forces but can be navigated efficiently should a player decide to do so. Urban terrain is difficult to navigate for the large corps sized units due to dense road networks within the city limits. Clear terrain is the most prominent terrain on the map and relatively easy to navigate while limited roads representing major highways provide an expedited mode of travel.

### 3.2.4 Models

Any tool designed to introduce officers to operational-level MCO must be equally rooted in joint theory and concepts. A tremendous amount of evolving literature exists defining these concepts. For successful game development, one must understand these concepts as they pertain to game design. WARLOC models the joint functions of movement/maneuver, joint fires, and sustainment along with combat through a series of supporting models.

**Maneuver, Defensive Counterair, and Logistic Models**

The movement/maneuver model is a straightforward hex movement system that is common in most turn-based wargames such as *Next War: Korea* and *Drive on Metz*. Mobile air and ground forces receive a movement allowance of movement points that represents the distance a unit could move over a given terrain in a 24-hour period of time [23, loc. 547]. The terrain effects chart lists the number of movement points required to enter a hex with that particular terrain type. Players spend movement points from their movement allowance for each hex they move into. If a unit does not have sufficient remaining points from their allowance, they may not enter that hex. This is typically followed by direct fire combat, as discussed in the combat modeling portion of this section.

WARLOC models the active air and missile defense portion of the defensive counterair mission. It models both air and missile targets and an active IADS. Vulnerable targets susceptible to attack include missile defense targets such as ballistic missiles and infrastructure such as air bases, logistic hubs, crossroads, sea ports and C2 nodes. The IADS consist of a combination of SAMs, antiaircraft artillery (AAA), and their supporting radar detection systems. In order to reduce complexity and gameplay length, WARLOC only depicts a limited number of SAMs systems and not AAA or extended radar detection. These systems have a radial range of one hex which is the same length as a standard ZOC. This limited range means that players must deploy the air defene artillery (ADA) systems in a point defense fashion where they can defend a single unprotected target. There are intentionally more targets than ADA systems which force players to prioritize which critical assets to defend. Air defense combat is resolved through a surface-to-air CRT as discussed in the combat modeling portion of this section.

Sustainment is modeled both directly, by representing air bases/logistic hubs and their distribution system, and indirectly, through a unit's combat power and movement allowance. These base/hub and distribution systems are vulnerable to attack from both TBM attacks and air interdiction. If the attack is successful, the defending player receives a penalty to their ground and air capabilities. This penalty is dependent upon the type of infrastructure attacked. A successful attack on a logistic hub slows distribution of supplies like fuel and ammo thus reducing units mobility and combat power. A successful attack on an airbase results in a loss of available sorties during the defender's next turn. Targetable infrastructure have a base defensive value which is increased when an ADA counter is in an adjacent hex. The added defense provided by ADA systems significantly decreases the probability of the attacker's success. In reality, the effect of a successful attack on an operational logistic target would not be felt for at least several days. However, in order to emphasize the cause and effect of logistic attacks in WARLOC, logistic impacts occur the very next turn.

### Joint Fires and Offensive Counterair

Joint fires is a substantial portion of WARLOC and is represented through the combination of air and land attack models that support the joint fires portion of the third learning objective (General Understanding of the Joint Function of Fire). Players target and leverage both land and air-based fires to achieve their desired effects. The land attack model takes into account indirect fire from both cannon and rocket systems. The cannon artillery has a range of 10 hexes and attacks enemy units while the TBM systems have a range of 25 hexes and attack enemy infrastructure (e.g., C2 nodes, logistic hubs and distribution networks, airbases, and seaports). The cannon artillery supports the players integration of tactical fires and maneuver while the TBM familiarizes players with operational fires.

The offensive counterair model directly contributes to the fourth learning objective (Introductory Experience with Air Power in a Joint Task Force) while the combination of the counterair and land models support the third learning objective. The goal of the air attack model is to help players understand the basic type of air operations. Players are able to plan and leverage these operations against an opponent with both deliberate and dynamic targeting. The joint targeting cycle and the air tasking cycles discussed in 2.2.3 are combined into a two-phased model; the air planning phase and the air operations-force execution phase. The air planning phase aggregates the entire joint targeting cycle (except for the assessment phase) and the first four stages of the air tasking cycle (determining the objectives, target development, and force allocation/air apportionment) [19] into a single phase. Players select a potential target to attack and assign a fighter squadron to a particular counterair/counterland role on an ATO (see Appendix C) menu, a design inspired by Joseph Miranda's Cyberwar XXI [63]. The target may be defended by ADA and defensive counterair patrols so the attacking player must dedicate aircraft sorties to counter these defenses. Ideally, players will want to first achieve air superiority thus ensuring freedom to conduct further interdiction and CAS operations. The player's plan, as it is laid out on the ATO is not used until the following turn.

The air operations-force execution phase represents the air tasking cycle's execution planning and assessment phases. The players execute their originally conceived attack plan using air assets in their assigned role against an appropriate target (deliberate targeting). For example, a fighter squadron assigned to a SEAD mission will attack an enemy's ADA system. The phase also accounts for dynamic targets that were not originally foreseen during the air planning phase. The player may change the air asset's role with a penalty to the air asset's combat power. For example, a target of opportunity appears and the player reassigns one of their fighter squadrons from a combat air patrol mission to an air interdiction mission in order to engage the target. Since the sortie was initially planned for a certain type of mission, their weapons load out matches a specific configuration. A short notice change to the sorties mission results in a loss of efficiency since the aircraft may not have the optimal load out for the new mission. The game penalty of a negative DRM is an abstract representation of this efficiency loss.

### Combat Modeling

WARLOC uses the combination of a CRT and a random number generator (simulating a dice roll) to determine the results of combat. The CRT lists the possible outcomes of a combat scenario. These outcomes are assigned a dice roll number based on the probability of that outcome occurring given the difference between the two combatants' combat powers (known as a combat differential). The combat differential, in general, is determined by subtracting the defending unit's combat power from the attacking unit's combat power.

WARLOC models four types of combat; direct fire, cannon artillery indirect fire, surface-to-air missile combat, and air-to-ground combat. Direct fire is conducted between two ground forces that are in each other's ZOCs. Indirect fire is conducted between a cannon artillery unit and any opposing ground unit. Surface-to-air missile combat occurs when an ADA engages either a missile or aircraft. In a TBM attack, an exception to the combat differential formula occurs. The defending ADA system's combat power is combined with the defensive value of the infrastructure it is defending. The missile's combat power is subtracted the ADA's combined defensive combat power. This is the opposite of traditional differential calculations in that the attacker's combat power is subtracted from the defender's combat power where normally it is the other way around. This is just one example of the complexities in modeling across two physical domains such as land and air.

The gameplay sequence is another attempt to minimize the complexity of interacting between the two domains by specifing the order of air combat resolution. It is the most complicated element in the entire game! The air combat sequence is air-to-air combat (Fighter Sweep & Fighter Escort) which uses the direct fire CRT. Next are either deliberate SEAD attacks using the air-to-ground CRT or dynamic SEAD attacks where a fighter escort attacks an ADA site. Dynamic SEAD attacks conducted by fighter escorts are conducted using the escort's SEAD value with a -2 DRM. Surface-to-air combat follows, where ADA systems engage incoming aircraft or missiles using the surface-to-air CRT. Lastly, air-to-ground combat (attack ops, air interdiction, CAS) is resolved using the air-to-ground CRT.

### 3.2.5 Rules

Rules create boundaries for players by limiting what actions they may take within the context of the game. They define how the models and database interact in relation to the scenario. Through the interaction of these three components, the wargame begins to take shape presenting itself as a system wargame as opposed to a seminar wargame.

Seminar wargames utilize an umpire, or adjudicator, who is the only one that knows how the rules, models, and database interact. With this knowledge the umpire moderates what is often a group discussion between players. Through this discussion the moderator presents scenarios, enforces rules, and maintains both the pace of the game as well as the engagement of the players.

WARLOC is a system wargame in which the combination of the map, database, models, and rules form a system. Players who decide to play the game agree the system is a fair determination of knowledge and performance within the scenario. By playing the game from start to finish the system provides the players insight, incites competition to maintain engagement, and ultimately provides a victor in the end.

### 3.2.6 Players

WARLOC is targeted toward Army captains and junior field grade officers with limited to no experience with joint operations. These officers often have little personal time to dedicate toward professional development let alone gameplay. This is one of the driving reasons to develop a more casual wargame with shortened gameplay.

## 3.3 Design Summary

We designed WARLOC with the goal of preparing officers to serve in a joint billet at an operational command. WARLOC is a casual wargame where the players achieve the game's learning objectives through active participation as the commander of a joint operational command. The level of combat presented in the game is one in which most junior officers have little to no personal experience. It attempts to reduce complex decisions and real-world processes into smaller and more manageable steps. The game introduces the players to air operations in a simplified format which gives the players a basic concept of airpower. Lastly, by using a turn-based gameplay system, the game provides players with the time to assess the operational scenario and then plan and execute their response rather than simply reacting without planning. This transition into deliberate planning and execution is a hallmark of an operational command. The next step after designing the wargame is to develop the wargame into a playable, web-based prototype.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 4:
# WARLOC Development

The production of the WARLOC prototype involved both designing and developing the wargame. Perla describes wargame development as the process which turns the initial design into a “fully fleshed-out, and hopefully, playable game” [4, loc 4843]. This development process confirms that the model, database, and scenario match their designed intent through multiple playtests, refinements, and rulebook updates.

## 4.1 Foundations

Before creating the web-based prototype a paper version of WARLOC was developed to ensure the playability of the basic wargame design. Next, the team adapted the paper wargame into a VASSAL module to assist with gameplay validation and prototyping possible component and user interface designs. Lastly, the team created a code base for the web prototype and tested it on several different platforms. This chapter discusses each of these steps in further detail.

### 4.1.1 Paper Wargame Development

The paper wargame components include the map, counters, CRTs, and rule set. Google Drawings was used to create the counters, Google Sheets to create the CRTs, and Google Docs to create the rule set. Chat, commenting, and the simultaneous editing features offered by the Google Docs suite allowed us to collaboratively develop the appearance and content of each component. Adobe Kuler, a free to use color selection tool that suggests matching complementary colors, was utilized to assist with the graphic design of the counters and player aid cards. The team discovered that Google Drawings supported the creation of basic components but lacked advanced image manipulation features such as transparency and color manipulation. This limitation particularly affected the development of the counters. In response, the team used GIMP to edit the counter images. Lastly, the team used HexDraw (one of the few non-free tools utilized) to create the game map.

Following the creation of the manual components, several rounds of playtesting were conducted. During each playtest meticulous notes were taken to capture and resolve issues.

The results of these playtests led to refinements in the rule set, counters, models, and CRTs. Playtesting a paper version of a wargame proved to be both time and resource intensive so a VASSAL module was developed to streamline remaining playtest requirements.

### 4.1.2 VASSAL Module Development

The WARLOC VASSAL module itself is a “compressed set of graphic, text, and other files, as well a descriptive file (called a Build File)” [64, p.10]. The WARLOC module contains the rulebook, all the graphics (map, counters, etc.), CRTs,ATOs, terrain effects chart, and all other game components created in the paper version. The VASSAL module recreated the experience of playing the paper version on a computer. WARLOC’s VASSAL module, as discussed in 2.6.6, does not include any game mechanics or automation. For example, initiating combat between North and South Korean unit counters requires the use of physical dice or a random number generator provided by the module. The value provided by the dice roll or number generator must then be manually referenced in the appropriate CRTs to determine the combat resolution. Once the combat result is determined, the required actions are then manually executed. If the combat resolution determines that either the defender or attacker were eliminated, the digital unit counters have to be explicitly flipped if step-lossed or moved to the graveyard if destroyed. Any discrepancies between players when playing the WARLOC VASSAL module must be resolved by referencing the WARLOC rulebook which is digitally provided within the VASSAL module.

VASSAL makes it simple to create a user friendly interface to play digital board games. It offers a clear illustration of how the user interface for any computerized or web-based wargame could be implemented. However, its lack of automation highlights the basic functionality needed in a web-based version to improve usability for the more casual player. Manually resolving combat, flipping counters, and measuring distances between hexes for movement or combat are easily automated and are the first of many steps in making a web-based wargame appeal to a military officer with limited free time.

## 4.2 WARLOC Game Design Document

Most commercial video games are developed using a GDD which includes descriptions of a game’s structure, level design, characters, user interface, artwork, etc. This is typically a written document that is distributed between the various design and development teams working on a game. The WARLOC VASSAL module acted as a basic GDD for the development of the web-based version. The module contained all the graphics (maps, unit counters, etc.) and all the data (CRTs, terrain effects, unit attributes, etc.) needed to build the actual web-based version of WARLOC. It provided a sense of how potential users would interact with the game's interface. Experiencing the game through the limitations of VASSAL helped to identify the tedious tasks associated with paper wargames that would need to be automated if it is going to be accepted by the largest audience possible.

The VASSAL module aided in separating the easy tasks from the more difficult. Automating the flipping and destruction of counters would be simple and straightforward while creating multiple windows for the player's ATOs and game map, and the interaction between the two, would clearly be the most difficult aspect to develop. Identifying these functions in the VASSAL module outlined the code development priorities for the WARLOC prototype. A working web-based version of the ground combat component of WARLOC's original design would be developed first with air combat added second, time permitting.

## 4.3 Web-Based Instantiation

The advantage of turning a paper wargame into a web-based wargame is the ability to automate certain tasks. Paper wargames often serve a niche community that is interested in extreme detail and does not mind the tedious tasks associated with playing the games. These tasks can include using CRTs to resolve combat, comparing a terrain effects chart to manually determine all available movement options, or determining whether or not a target is within range of an indirect fire capability. The automation in WARLOC was developed from a combination of the paper version's rule set as well as the goal to improve usability. We believe that removing many of the tedious tasks of paper wargames will make a web-based version more appealing to a wider range of players.

Code development on the web-based version was conducted using the VASSAL module as the GDD and Brackets as the code editor. Interaction between the the game map, hex grid and player was achieved first. Next, both the unit counters and hexes were built as JavaScript objects. With the map, hexes and counters existing in the browser, the game logic was developed automating a number of the tedious tasks conducted in the paper version of WARLOC.

### 4.3.1 Location Reference System

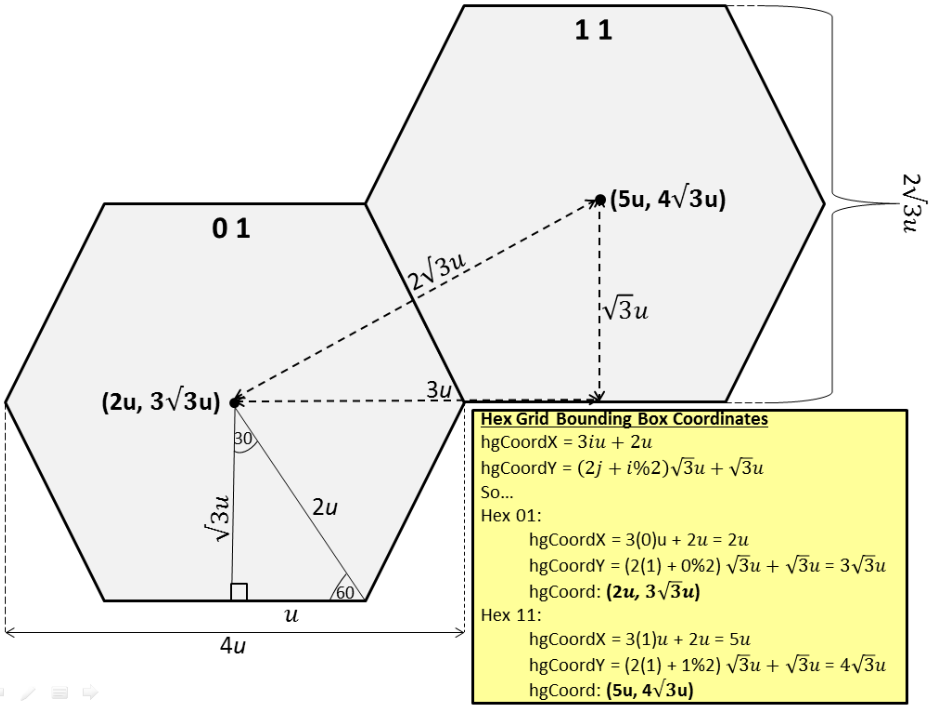
WARLOC's gameplay centers around its map where users plan and execute their strategies through direct interaction. For performance and simplicity sake, WARLOC loads a premade map image, created in HexDraw, and executes graphics code to superimpose a hex grid over the map image. The grid is independent of the map image and is comprised of 25 columns and 29 rows totaling 725 hexes of equal size. Though the game map is the center of the game, the hex grid is the center of interaction making the generation of the hex grid and corresponding hex selection functions significant tasks in the game's development.

#### Hex Grid Generation

Web browsers utilize the "<canvas>" element to allow developers to write graphics code. Developers can create both two-dimensional and three-dimensional worlds in the browser window by executing the graphics code every frame. The hex map in WARLOC is actually an amalgamation of several different objects, coordinate systems, and their associated supporting functions. Though libraries like EaselJS include a function for drawing hexes, it can be time consuming to learn a brand new JavaScript library to use it effectively. It is for this reason that we elected to develop WARLOC using core JavaScript and basic hex geometry to generate the hex grid.

The paper version of WARLOC's Korean map image serves as the base layer of the game on which we overlayed an independent hex grid. A bounding box was developed to surround the game map image with an origin located in the bottom left corner of the of the browser window. Using this origin, all the hexes were created using a nested loop with the number of columns (25) and rows (29) acting as the stopping condition. A double array contained the hex objects whose index numbers match the hexes' column and row locations. For example, Hex 01 can be referenced as mapHexes[0][1].

JavaScript only requires the coordinates of the vertices in order to draw each side of the hex. Knowing the center point and side length is all the information needed to draw the entire grid. Using the bounding box origin as the starting point, Figure 4.1 illustrates how the center point coordinates of each hex can be determined knowing only the side length (*u*) and the column (*i*) and row (*j*) position of the hex. This information was used to generate the entire hex grid.



**Hex Grid Bounding Box Coordinates**
hgCoordX = $3iu + 2u$
hgCoordY = $(2j + i\%2)\sqrt{3}u + \sqrt{3}u$
So…
Hex 01:
hgCoordX = 3(0)u + $2u$ = $2u$
hgCoordY = (2(1) + 0%2) $\sqrt{3}u + \sqrt{3}u$ = $3\sqrt{3}u$
hgCoord: **($2u$, $3\sqrt{3}u$)**
Hex 11:
hgCoordX = 3(1)$u$ + $2u$ = $5u$
hgCoordY = (2(1) + 1%2) $\sqrt{3}u + \sqrt{3}u$ = $4\sqrt{3}u$
hgCoord: **(5u, 4√3u)**

Figure 4.1: Hex Geometry.

Using the pre-made map image (see Appendix F with the hex grid drawn on top improves the game's performance but also creates other challenges. The greatest challenge is syncing the canvas coordinates with the hex grid coordinates. For example, since the coordinate, $(2u, 3\sqrt{3u})$ is in reference to the bounding box origin, it will not appear where one might expect when drawn on the canvas element. The bounding box hex grid coordinates must be translated into canvas coordinates. Figure 4.2 depicts the three coordinate systems needed to complete this translation.

To properly draw the hex grid, a conversion from hex grid coordinates, to math coordinates, to canvas coordinates is required. Math coordinates, the traditional $(x, y)$ coordinate system with which most people are familiar, are only an intermediary step to translate the hex coordinates into canvas coordinates. Canvas coordinates are almost identical to math coordinates except that the screen consists of negative $y$ coordinates with the positive $y$ coordinates existing off screen .

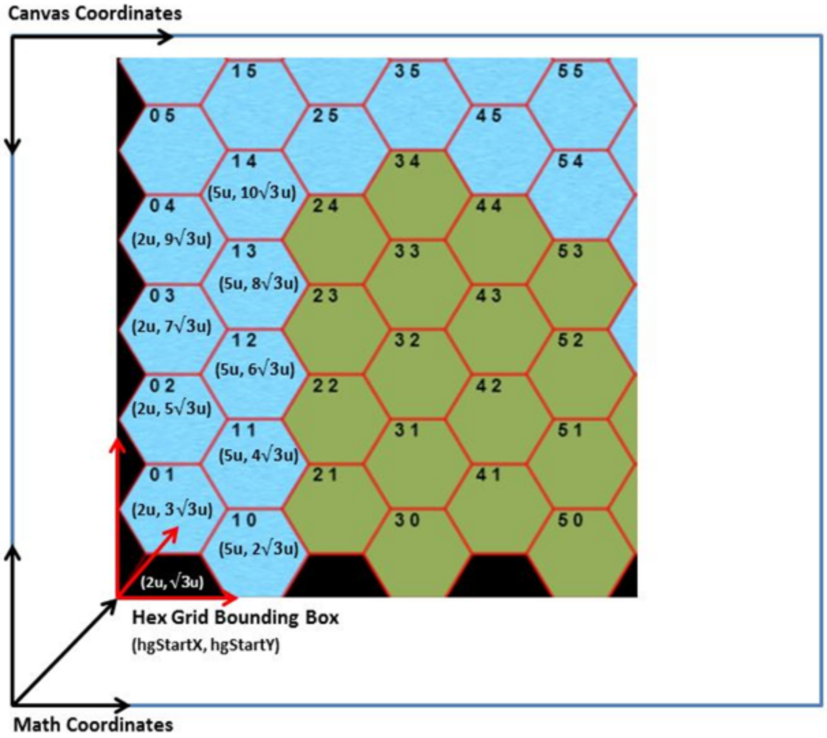
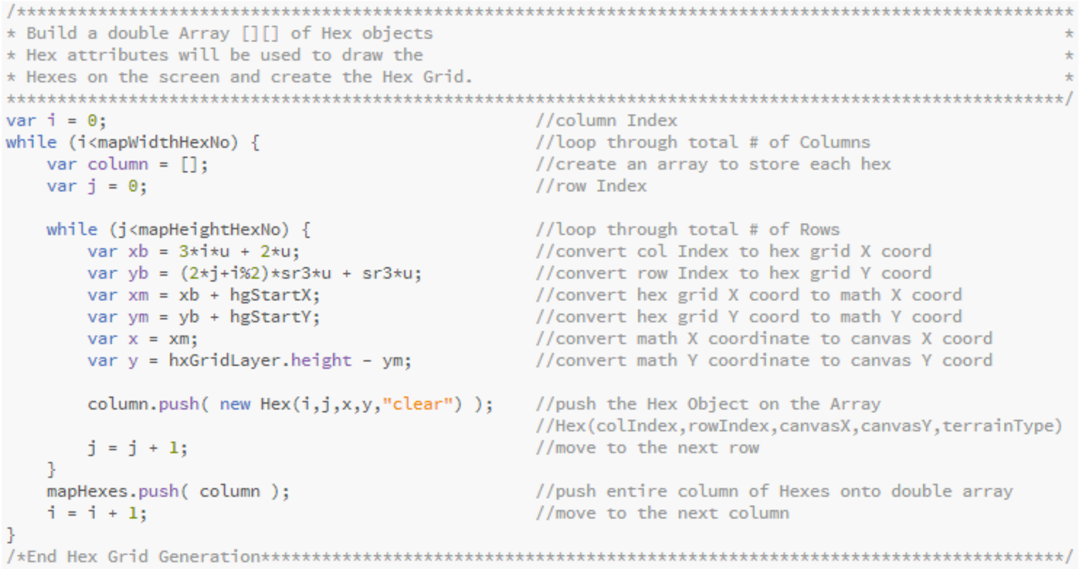


Figure 4.2: WARLOC Coordinate Systems

Each hex was created as an object and assigned a column and row index within the hex grid coordinate system as well as appropriate canvas coordinates for the center of the hex. These indices and coordinates are used to identify the hex objects and draw them on the screen, respectively. The code to create these hex objects with the appropriate canvas coordinates is depicted in Figure 4.3. The code is located in the game setup function so the global variables (side length ($u$), $\sqrt{3}$ (*sqr3*), and the bounding box origin *hgStartX* and *hgStartY*) are defined elsewhere in the code.



```
/*********************************************************************************************************
* Build a double Array [][] of Hex objects                                                               *
* Hex attributes will be used to draw the                                                                *
* Hexes on the screen and create the Hex Grid.                                                           *
*********************************************************************************************************/
var i = 0;                                          //column Index
while (i<mapWidthHexNo) {                           //loop through total # of Columns
    var column = [];                                //create an array to store each hex
    var j = 0;                                      //row Index

    while (j<mapHeightHexNo) {                      //loop through total # of Rows
        var xb = 3*i*u + 2*u;                       //convert col Index to hex grid X coord
        var yb = (2*j+i%2)*sr3*u + sr3*u;           //convert row Index to hex grid Y coord
        var xm = xb + hgStartX;                     //convert hex grid X coord to math X coord
        var ym = yb + hgStartY;                     //convert hex grid Y coord to math Y coord
        var x = xm;                                 //convert math X coordinate to canvas X coord
        var y = hxGridLayer.height - ym;            //convert math Y coordinate to canvas Y coord

        column.push( new Hex(i,j,x,y,"clear") );    //push the Hex Object on the Array
                                                    //Hex(colIndex,rowIndex,canvasX,canvasY,terrainType)
        j = j + 1;                                  //move to the next row
    }
    mapHexes.push( column );                        //push entire column of Hexes onto double array
    i = i + 1;                                      //move to the next column
}
/*End Hex Grid Generation********************************************************************************/
```

Figure 4.3: WARLOC Hex Grid Code

The hex objects contain a function that will utilize their *x*,*y* variables to draw the hex on the screen. With the hex objects using canvas coordinates, it only takes small adjustments to the side length of each hex to properly overlay the grid on the map.

### Hex Selection

To translate from a user's click to selecting the correct hex on the grid requires the program to reverse the translations discussed in hex grid generation. An outline of this process is as follows:

1. User clicks on game map, canvas coordinates are returned.
2. Convert canvas coordinates to math coordinates.
3. Convert math coordinates to hex grid bounding box coordinates
4. Convert hex grid bounding box coordinates to rectangular column and row indices
5. Convert rectangular column and row indices to hex column and row indices

Steps 1-3 are completed by reversing the process discussed in hex grid generation. In order to determine whether or not the player clicked anywhere within the area of the hex, the bounding box is conceptually broken into a rectangular grid as see in Figure 4.4.

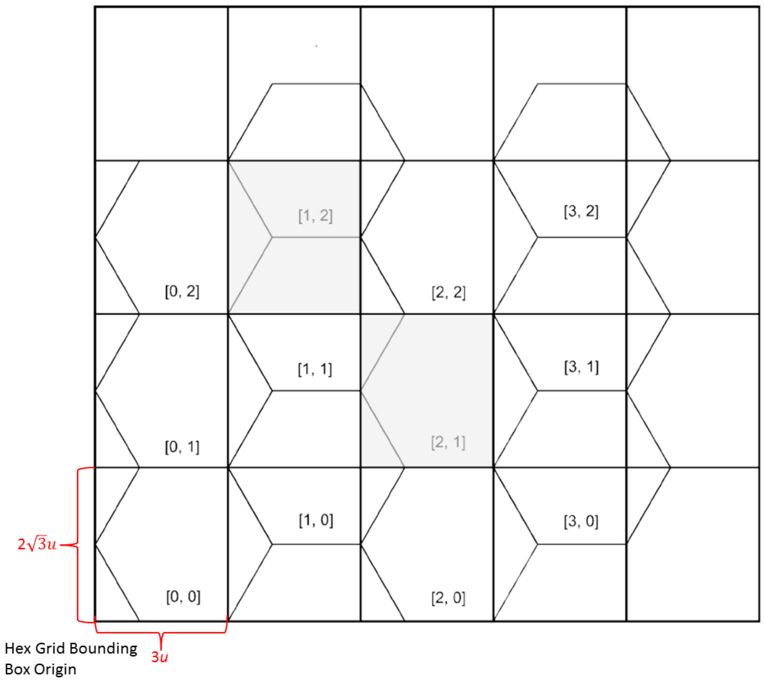


Figure 4.4: Rectangular Grid Overlay

Each rectangle has a length of $2\sqrt{3}$ and a width of $3u$ corresponding to the length and width of the hexes. Each rectangle is occupied by at most three different hexes. Dividing the hex grid bounding box coordinates by the width and height of the rectangle provides the column and row indices of the rectangle as seen in equations (4.1) and (4.2):

$$\text{Rectangle Column} = \lfloor \frac{\text{hex grid bounding box X coordinate}}{3u} \rfloor \tag{4.1}$$

$$\text{Rectangle Row} = \lfloor \frac{\text{hex grid bounding box Y coordinate}}{2\sqrt{3u}} \rfloor \tag{4.2}$$

Since the hex column and rows are offset, two possible conditions exist within any rectangle. These conditions are based on whether the rectangle column index is odd or even as seen in Figure 4.5.

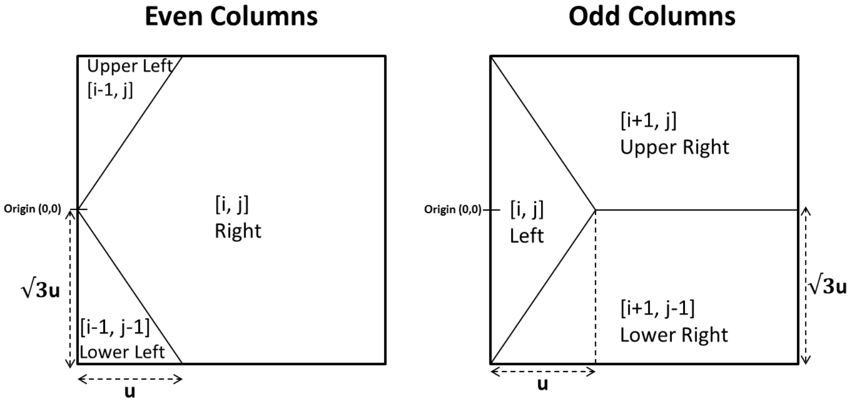


Figure 4.5: Odd and Even Grid Squares

The mouse click, currently represented as hex grid bounding box coordinates (step 3), is converted into rectangle coordinates (step 4) using equations (4.3) and (4.4)

$$xr = xb - 3u(\text{Rectangle Column}) \tag{4.3}$$

$$yr = yb - 2\sqrt{3u}(\text{Rectangle Row}) - \sqrt{3u} \tag{4.4}$$

$\sqrt{3u}$ is subtracted from the y coordinate since the center of the left side of the rectangle is treated as the origin of this new coordinate system. These new rectangle coordinates, $(xr, yr)$ indicate the position the mouse click occurred within the rectangle. The region that the click occurred is determined using the following:

If the rectangle column index is even

$$\text{If:} \quad xr > \frac{|yr|}{\sqrt{3}} = \text{Right}, \quad \text{Else:} \quad yr > 0 = \text{Upper Left}, \quad \text{Else: Lower Left} \tag{4.5}$$

If the rectangle column index is odd:

$$\text{If:} \quad xr < u - \frac{|yr|}{\sqrt{3}} = \text{Left}, \quad \text{Else:} \quad yr > 0 = \text{Upper Right}, \quad \text{Else: Lower Right} \tag{4.6}$$

Once the region is determined, the rectangle column and row indices $(i, j)$ are modified as depicted in Figure 4.5. These values are used with the double array of hex objects and the selected hex is obtained (step 5). WARLOC's pickHex function can be seen in detail in Appendix G.

### 4.3.2 Unit and Hex Objects

The main components of WARLOC include the game map, hexes, and unit counters. Both the counter and hex objects contain numerous attributes and functions ranging from a unit's combat power to a hex's terrain type. JavaScript objects are key-value pairs allowing programmers to create objects which contain values of any type such as primitives (numbers, strings, booleans), functions, or other JavaScript objects. These values can easily be referenced by simply knowing the variable name that contains the value.

The hex objects themselves consist of ten attributes and three functions. As discussed in 4.3.1, each hex contains two attributes representing their center point in canvas coordinates. This center point is used by the hex object's "drawHex()" function to actually draw the hex on the screen. The structure of the hex object is depicted in Figure 4.6.

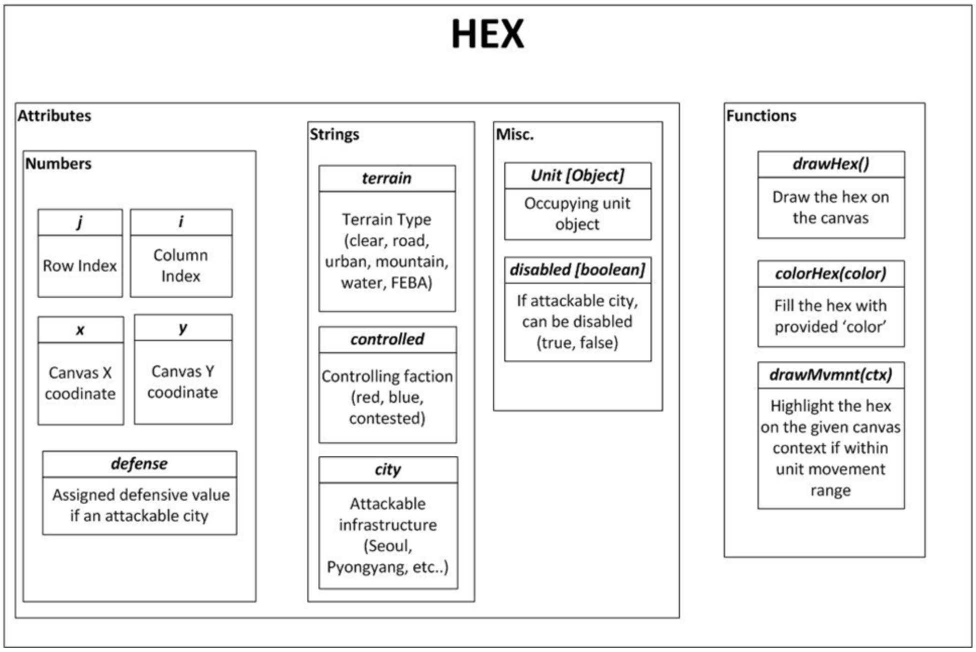

HEX
Attributes
Numbers
j
Row Index
i
Column Index
x
Canvas X coodinate
y
Canvas Y coodinate
defense
Assigned defensive value if an attackable city
Strings
terrain
Terrain Type (clear, road, urban, mountain, water, FEBA)
controlled
Controlling faction (red, blue, contested)
city
Attackable infrastructure (Seoul, Pyongyang, etc..)
Misc.
Unit [Object]
Occupying unit object
disabled [boolean]
If attackable city, can be disabled (true, false)
Functions
drawHex()
Draw the hex on the canvas
colorHex(color)
Fill the hex with provided 'color'
drawMvmnt(ctx)
Highlight the hex on the given canvas context if within unit movement range

Figure 4.6: The Hex Object

The unit objects are the largest class of objects used in the game. Each counter has two overarching states, full power and step-lossed. All unit objects begin the game at full power and enter the step-lossed state only after receiving damage. This requires the unit objects to maintain two values for almost every attribute to include, combat power, movement, and even the counter image. Using the single boolean attribute "dmgd" makes it easy to determine the state of the unit and which attribute is needed. The *engage()* function is an excellent example of this as it uses the unit's "dmgd" attribute to return the appropriate combat power value, "cp" or step-lossed compat power ("slcp"). The unit counter object is depicted in Figure 4.7.

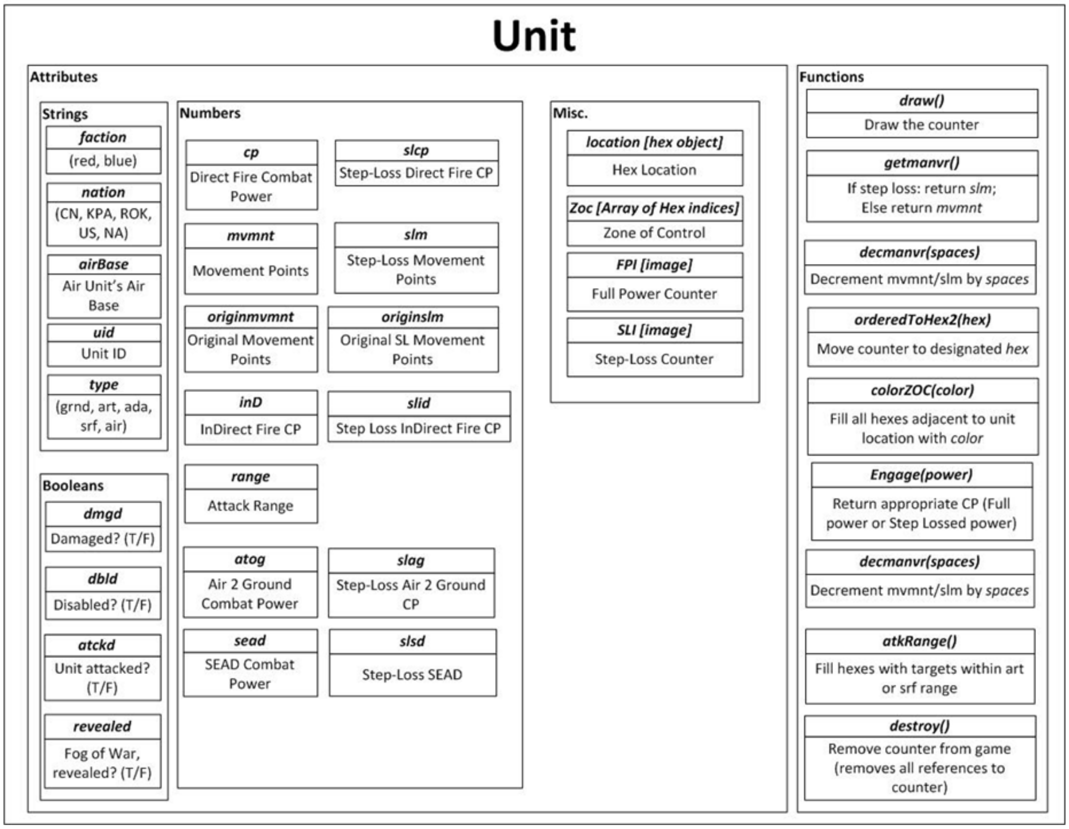


Figure 4.7: The Unit Object

### 4.3.3 Adjudication

Paper based wargames and VASSAL modules require players to manually resolve combat through the use of dice and a CRT. Automating this task is essential in simplifying game play for the web-based version. The logic behind automating some of these tasks can be seen in Appendix H. The CRTs are hardcoded into WARLOC using arrays matching the dimensions of the CRTs. These arrays contain all the combat results found in WARLOC's CRTs (attacker eliminated (AE), defender eliminated (DE), attacker retreat (AR), defender retreat (DR), equal elimination (EE), no effect (NE), disrupted (DT)). When combat is initiated, JavaScript's random number generator is used to replicate the die or dice roll. The generated value (slightly altered to reflect the fact that arrays start at position 0) is then automatically referenced within the appropriate CRT array and a combat result is obtained. This result is passed to a function that uses it to apply the effects of the combat result. For instance, if the combat result is DE, the defending unit counter's "dmgd" attribute is set to "true". As discussed in 4.3.2, the "dmgd" attribute causes the program to use the unit object's step-lossed image and refer to its step-lossed combat power, movement values, etc., for the remainder of the game. If the unit is already damaged, the units "destroy()" function is called and the counter is removed from the game. The players receive notification of the combat result from a pop window and observe the game's state change as a result of the indicated combat result.

### 4.3.4 Movement

WARLOC's map consists of six different terrain types: clear, urban, road, water, mountain and FEBA. Automating the terrain data was attempted by trying to sample the map image at the center point of each hex object. This method proved ineffective as instances of black pixels were sampled from the map image's text and from the outlines of the mountain and water graphics. In the end, a terrain data file was created. This file contained manually entered terrain data in the form of an array that was used in the game's setup function to assign each hex its designated terrain data.

Each terrain type has different effects on the movement of the counters. Clear hexes cost two unit movement points to enter, mountains cost four, urban areas cost three, roads cost one, and water cannot be entered by any unit. The FEBA has the most unique functionality. Any ground unit counters with a faction attribute equal to "red" is affected by the FEBA. Entering the FEBA costs a red unit two movement points. Upon entering the FEBA the red counter will lose all remaining movement points leaving it unable to move until the following turn. In addition to this, any red units attacking out of the FEBA receive a -2

DRM. ROK-US counters are able to move freely through the FEBA and receive no DRMs when attacking out of or into the FEBA.

### Path Search Algorithm

In paper wargames, players have to manually calculate the terrain movement cost of individual hexes. The power of computers once again reduces the tediousness of paper wargames. In WARLOC, when a player selects a ground unit and clicks the maneuver button, all the possible locations for the unit to move are highlighted. A recursive algorithm, called *detMove*, takes the selected unit's movement points and determines all available movement options for the unit. This algorithm does not try to find the best path but finds all paths leaving it up to the player to decide where to move the counter. Pseudocode for the algorithm is provided in Appendix I.

Figure 4.8 depicts the game map after a Juche Alliance ground force has been selected and the maneuver button clicked. Even though the red counter has 10 movement points, its possible moves are slightly restricted. FEBA hexes drain all remaining movement points if entered so no hexes North of the FEBA are filled. The red unit's back is against the water which cannot be entered by any ground unit so they are not filled with a color and are not a movement option. Terrain to the South has been filled with the color gray so it is a safe option but takes the red counter away from its goal, Seoul. Finally, ROK-US forces' ZOCs stop the red unit from passing through hexes adjacent to their forces. These hexes are identified by their blue and yellow coloring. The hexes behind the ROK-US ZOCs are not filled in since they are not accessible. This figure displays how units distributed across the terrain can restrict the enemy's freedom of maneuver by being thoughtfully placed. Unfortunately, the ROK-US forces have left an opening to Seoul's West. The Juche Alliance player has a clear path into Seoul in this scenario which will result in a North Korea victory.

The *detMove* algorithm has been used in all playtests conducted up to this point. The algorithm fails to handle road hexes perfectly. It treats adjacent hexes containing roads as costing one movement point to move between them. This is accurate when the roads are connected and are going in the same direction but not if two adjacent hexes contain two parallel or unjoined roads. This is due to the fact that the roads are part of the map image and not part of the hex objects. The hex objects could have some added attributes manually defined to determine what direction the road is facing but this was not attempted due to

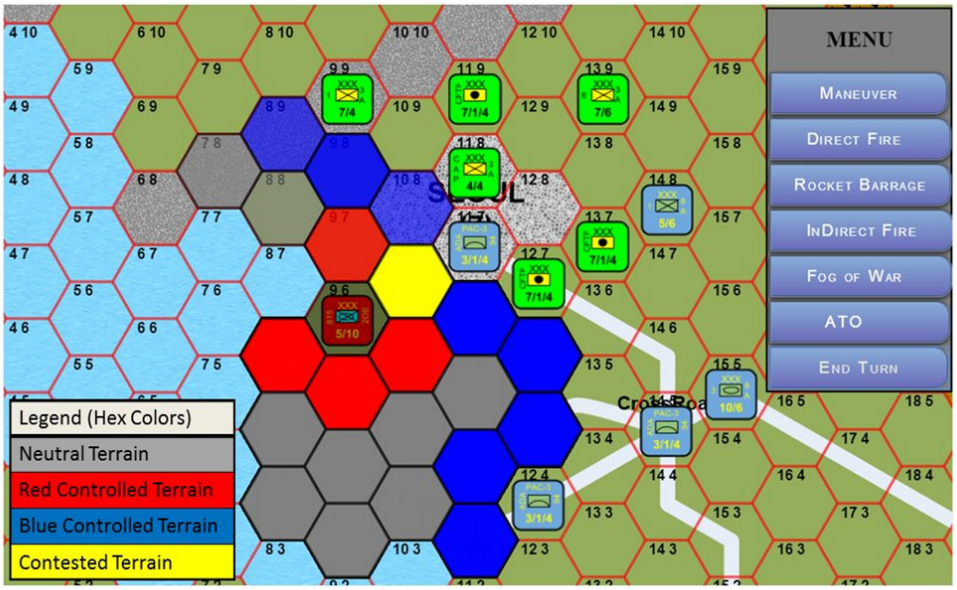


Figure 4.8: Display of *detMove* Algorithm Path Finding

time constraints.

Players can currently exploit *detMove's* road loophole and move a unit slightly faster than technically allowed by traveling through adjacent hexes that contain a road facing any direction. Since the number of roads are minimal and do not exist in the highly contended zones, e.g., South of Nampho and North of Seoul, the advantages this loophole provides are negligible. Regardless, fixing this algorithm or adding extra attributes to the hex objects should be a priority for any future work.

### 4.3.5 User Interface

We initially envisioned a user interface capable of guiding players through the joint functions in hopes to reinforce the joint function familiarization learning objective of the game. A very early conceptual prototype of WARLOC's interface included the use of a CSS flexible box layout [65]. The CSS flexible box provides an adaptable layout that can be used to create a simple interface. Buttons and windows are capable of changing size as screen real estate dictates. Unfortunately, the code used to generate this layout proved to be more complex than anticipated. Our lack of experience with CSS and lack of time made using the flexible box more difficult than originally anticipated and began taking time away from the game development. The code to develop the layout in Figure 4.9 is open source and thus free to use. Even though we moved from the flexible box design we feel that it is a useful tool and warranted a discussion in this thesis.

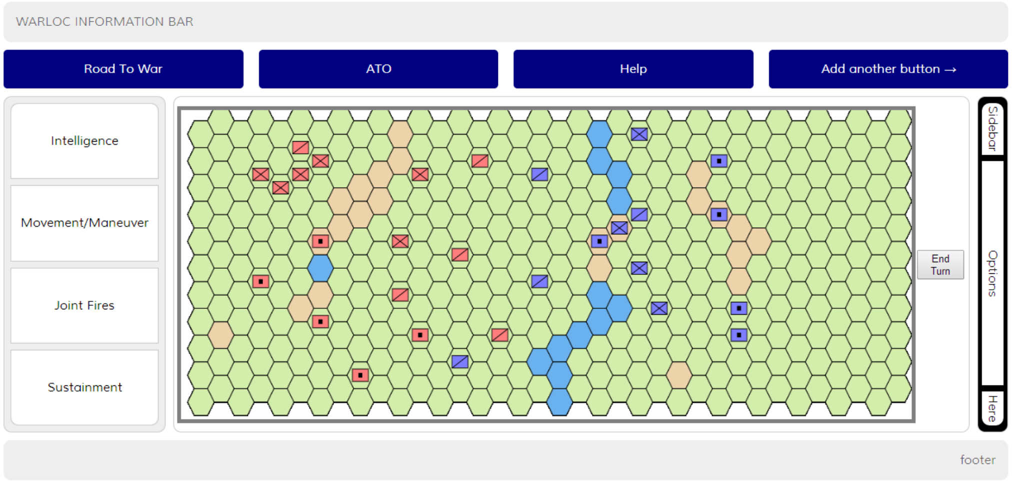


Figure 4.9: Original User Interface Concept, flexible box code from [66]

Electing not to use the CSS flexible box did not remove the requirement of a user interface for our game. WARLOC's current interface was designed with mobile devices in mind making the game highly dependent on the mouse click rather than a keyboard. The team developed the prototype to work primarily in a desktop web browser leaving mobile platform optimization for future development. Figure 4.10 shows the user interface implemented within the working prototype. A larger screen shot is provided in Appendix J.

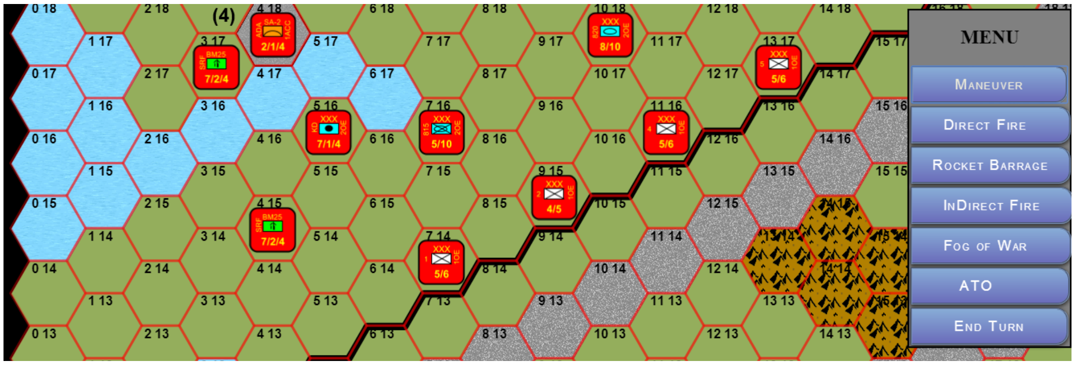


Figure 4.10: WARLOC User Interface, button code from [67]

In the prototype, players click on a unit to select it. To execute any of the available functions of the unit, the appropriate menu button must be clicked. Clicking the menu button causes the map to highlight all executable options on the map related to the button's function. If the player decides they do not like their options they can click on a non-highlighted hex resulting in the friendly unit being deselected. The player can then continue the game by selecting another counter or ending their turn.

Selecting a ground unit and clicking the maneuver button invokes the *detMove* function discussed in 4.3.4. If the direct fire button is clicked, the unit's ZOC is shaded gray while any enemy units within the ZOC are shaded red. In order to avoid oversaturation of the map for indirect fire attacks, WARLOC highlights only hexes that are within range and are occupied by enemy units. A similar technique is used for the Juche Alliance player's SRF though targetable infrastructure is highlighted as opposed to enemy units. In the game's rules, as can be expected, only artillery units and SRF units can conduct indirect fire and rocket attacks respectively. To ensure this is the case in the game, WARLOC informs the player if they attempt an illegal move or make a mistake and does not execute the selected action. As it stands now, WARLOC does not offer an undo button so any actions executed in the game are irreversible.

The fog of war button prevents a player from viewing their opponent's counters during his or her turn. The player can only reveal enemy counters by entering the enemy counter's ZOC. Once revealed, the unit is no longer hidden and is viewable by both players until the end of the game. This feature can be turned off by clicking the menu button a second time. If this feature is turned on *detMove* will not depict who controls what terrain as discussed earlier in Figure 4.8. Instead, all movement options will be filled with the color gray. The ATO menu button is currently only a placeholder as WARLOC's air piece has yet to be implemented.

## 4.4 Game Development Summary

During the development phase, the initial wargame design was adapted over several different formats with the end goal of creating a web-based version, hence the title *Web-based* Army Repeatable Lesson in Operational Combat (WARLOC). The game went from concept to paper with a focus on creating the look and feel of the game as well as playability.

Following several rounds of playtesting and refinement, a VASSAL module was created providing insight on player interaction with a computer version of the game as well as streamlining the refinement process. Lastly, and most importantly, a prototype web-based version of the ground warfare portion of WARLOC was developed. This involved a number of techniques including creating a hex-based map system, instantiating unit counters, and developing a number of JavaScript functions to execute the wargame's models.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 5:
# Conclusion

The overarching intent of this thesis was to address a shortage of joint operational MCO experience among junior officers arriving to operational commands. Wargaming has historically been an excellent method used throughout history by military commanders to train their forces in the planning and operation of warfare. The drawback to these games is that they are often complex and time-consuming. Wargames must be more intuitive, approachable, and less time-consuming while still meeting design objectives. Web-based technology is one of the fastest growing technologies and offers an excellent venue for a game due to its accessibility and ease of distribution. Combining a casual wargame with this technology reduces the amount of time required to set up and play while also increasing the distribution to the audience.

The overall end state of this thesis was the development of a web-based wargame prototype which familiarizes players with MCO at the operational-level. This development included the creation of both a paper wargame and a VASSAL module as steps to creating the web version. WARLOC, as of the publication of this thesis, has achieved the goal of creating a prototype that facilitates the ground component gameplay. While the prototype is not complete, players can use the completed paper version and VASSAL module for the full learning experience. This chapter discusses our key findings in relationship to the thesis research questions, implications, potential future work, and limitations of this game's development.

## 5.1 Wargame Design Findings

The first two research questions address wargame learning objectives and design. The key to designing WARLOC was to make a game that captivates its players while familiarizing them with concepts they may never have experienced or even considered. WARLOC's design is based on MCO doctrinal concepts from both the joint force and individual service branches. These doctrinal concepts have been distilled into a system of models and a database that support the game's learning objectives. The objectives include familiarizing players with both joint and component-level task organizations as well as operational-level

MCO. Another objective is introducing the players to the joint functions of fire, movement/maneuver, and sustainment. Lastly, it gives the players an introductory experience with air planning and operations.

### 5.1.1 Immersive gameplay requires clear learning objectives, balanced gameplay, and a firm understanding of your audience

WARLOC, in its paper and VASSAL module form, is a casual wargame with a setup and gameplay time of approximately 2 hours. In its web-based form, playtests often took about an hour to play all eight rounds. WARLOC, in any form, has the potential to immerse its players during game play which results in intense focus on the game and its content and can engender repeated play. Immersion in repeated play of WARLOC places players in a MCO environment making MCO decisions. Through repeated play, officers can become familiar with the values of conducting joint operations which previously had to be gleaned from reading dry and unengaging joint doctrine text.

Balancing accuracy and simplicity in order to achieve immersive gameplay is very difficult. We abstracted a large number of real-world processes to try to make the game playable while still achieving the learning objective (e.g., air warfare). At times, there were significant second and third-order effects from these abstractions that forced us to slightly increase the complexity. To balance this complexity we had to make assumptions about our playing audience's familiarity with certain topics. For example, we assumed that the players would be familiar with tactical ground warfare so the maneuver portion of the game was simplified while more detail was added to the games air warfare model.

In the VASSAL module and paper version, the Juche Alliance forces outnumber acROK-US forces nearly 2:1 (8:5 to be exact) making the ROK-US forces the more challenging side to play. The web-based version's lack of air combat makes it difficult for ROK-US forces to conduct attacks behind enemy lines. Research has suggested that increased challenge can lead to increased retention, although too much challenge can lead to the opposite outcome [68]. This further suggests the importance of achieving balance in a wargame intended to familiarize its players with educational material.

### 5.1.2 Approachability of the wargaming community

Wargamers are passionate about wargames and are an extremely friendly and helpful group of fans. Similarly, designers and developers are very willing to assist any new game designers. Neither author had a great deal of wargaming experience so we reached out to the to wargaming community at the Pacificon Game Expo [69] for assistance. Face to face meetings with several designers led to countless hours discussing wargame design concepts and tips. This assistance reached beyond just gaming conventions. Several more designers, such as John Tiller and Gary Morgan, contributed enormous amounts of personal time through phone calls, chat sessions, and email conversations that guided WARLOC's design. Many of these designers have decades of experience and understand how to make games for a particular objective. A new designer/developer should reach out to this community for advice and assistance.

### 5.1.3 Joint operations balance

Complexity increases with each additional physical domain added to a wargame since both cross-domain (e.g., air vs ground) and intra-domain (e.g., air vs air) interactions must be modeled. Modeling these interactions can be difficult when trying to balance complexity and playability of a wargame. A wargame designer must achieve balanced combat between the service components mirroring a rock-paper-scissors-like balance where each component can win a conflict depending on the conditions.

One challenge is simply gauging combat power. WARLOC used a multi-operation combat power scale for the air counters. Each aircraft may engage both land and air targets with their air-to-air, SEAD, or air-to-ground combat powers against the target's direct fire combat power. These engagements, as modeled in WARLOC, resulted in air components that were too effective in certain situations and too ineffective in others. Balance was not achieved. One area of research that may help improve this balance is the measurement of effects that aircraft munitions have on ground and air targets.

Another challenge is gameplay speed. We wanted a full game of WARLOC to last less than two hours. WARLOC's play time is highly dependent on the time it takes each player to play through their respective turns. WARLOC playtests revealed that air combat was the largest time contributor to each player's turn. Though air warfare is a large portion of

WARLOC, it serves largely as a support function as players are required to successfully maneuver their ground forces to control Seoul and win the game. The length of time a player spends planning and executing the air portion versus the maneuver portion further suggests imbalances in the game. We want each joint function to feel equally important in a player's victory. One possible solution to reduce the air combat time requirement would be to reduce the total number of air counters in WARLOC.

Lastly, we wanted to add a special operations forces (SOF) component to WARLOC ever since the game's initial conceptual design but were concerned about added complexity and gameplay length. Near the end of the development cycle, a SOF component was added to the web-based version. Players cannot interact directly with this SOF component as it acts as a random disruptive effect against TBM and infrastructure. This additional feature was an attempt to balance the web-version due to the non-existent air component. This feature has not been thoroughly playtested so the overall impact is unknown at the time of this writing.

## 5.2 Wargame Development Findings

We developed a web-based prototype of the overall design allowing players to experience the ground combat contained within WARLOC. The game can be found at http://faculty.nps.edu/cjdarken/pavek-starken-2014. We fully believe that a game similar to our VASSAL module can be fully recreated as a web-based version. This is only a matter of using CSS to stylize the WARLOC's HTML and adding the needed JavaScript code for functionality. Neither author was intimately familiar with CSS so a large learning curve existed and lack of time prevented us from pursuing this level of development. Time also prevented full incorporation of the ATOs and thus the air component of WARLOC.

Despite WARLOC's imbalances, WARLOC has value. There is a sense of pride in achieving victory when playing as the ROK-US forces. This intrinsic reward can increase a motivated player's immersion which can lead to a flow-like state. Intense focus on a game rooted in joint doctrine offers another opportunity for officers to familiarize themselves with material they may not receiving in their current positions.

WARLOC uses, thus familiarizes players with, doctrinal operational symbols. As units and critical infrastructure become damaged, players begin to understand how these effects can influence a combat operation at the operational level. A step-lossed ROK-US counter, not receiving supplies due to the disruption caused at Daejeon by a Juche Alliance interdiction attack, may not be capable of moving even one hex. They begin to understand the true value of supply lines like major roads or intersections similar to the attackable crossroads found in WARLOC. Players realize the importance of protecting major ports like Pohang to ensure that reinforcements can arrive. These situations help officers understand the joint function of sustainment at the operational level of war.

With victory conditions based solely on whether or not a North Korean force enters Seoul, players begin to understand that complete destruction of the enemy is not always the direct path to mission accomplishment. The power of ZOCs reinforce the joint function of maneuver as a well-placed unit restricting movement of enemy counters may be all that is needed to achieve victory.

Players are familiarized with the joint function of fires through the use of cannon artillery, TBMs, and air sorties. Though somewhat abstracted, players are familiarized with the difficulty in planning and executing air operations. They are also familiarized with terminology associated with air operations, both of which will assist in any future joint assignment. The value of ADA is constantly reinforced in the player's minds as they realize no interdiction mission is successful without the proper support. Overall, players are familiarized with a host of strategies and considerations at the operational level that may not necessarily exist at all at the tactical level of warfare.

### 5.2.1 Rapid Prototyping via VASSAL

Moving to VASSAL was a decision that saved the most time in developing WARLOC. Initially, a significant amount of time and resources were expended repeatedly printing and cutting out unit counters every time a change was made. Through the use of VASSAL, changes could be made to the game's map or unit counters and a playtest conducted in minimal time. The added bonus of using VASSAL is that a working, manual, digital version of the game is produced. Though not web-based, creating a VASSAL module of a well designed wargame can be distributed more efficiently than a paper version. VASSAL offers a server to host any modules developed with the VASSSAL engine. Care should be taken when utilizing this option as any VASSAL modules uploaded to this server are made available to the entire VASSAL community. If the military were to decide to use the VASSAL engine to produce confidential digital versions of their strategic wargames, the VASSAL server should be avoided and another distribution system implemented.

### 5.2.2 Code Development Process

Design, play, refine was used to develop the paper version of the WARLOC but not so much for the web-version. We were intensely focused on perfecting the design of the paper version before beginning any code development. Time forced us to begin code development when it should have been forced by our development process. This hurt the end product as we did not have the time to explore the more difficult aspect of the game, WARLOC's air combat.

### 5.2.3 Some Libraries are Useful, None are Required

Time was spent researching the different available JavaScript libraries which were never used. Though these libraries can offer solutions to certain problems within a complex project, the time and effort needed to learn the intricacies of these libraries can quickly diminish any advantage they provide. In addition, these libraries are most often governed by one of the many different open source licenses. These licenses can cause unanticipated complications in releasing web applications to a limited audience such as the military community.

## 5.3 Implications

Two Naval Postgraduate School (NPS) students, with limited programming experience, produced the WARLOC prototype working part-time with marginal costs. This is in direct contrast to the expensive, resource-intensive simulation development typically found in the military. A dedicated game development team of military personnel could easily produce affordable, timely, and flexible web-based applications for the service branches.

WARLOC has the potential to supplement a lectured course in joint theory. Though it is debatable on how much one can learn by playing a game alone, empirical evidence does exist showing the effectiveness of games in the classroom. Richard Blunt, a retired naval officer and learning consultant, conducted three studies at a university related to using games in the classroom. He found that the classes that used games in their course significantly outperformed students in classes without the game [70]. Games provide hand on experience with the subject matter which can facilitate in class discussion.

## 5.4 Recommendations for Future Work

First and foremost WARLOC is a simulation, even if considered abstract. Creating any simulation accurately takes time and constant refinement. There are still issues with the overall game balance to include accurate unit combat power values, adequate CRTs, and the ability of each side to equally interdict each other. Extensive playtesting will easily improve these factors but require the one resource we ran out of, time.

The game is operational in the web browser of desktop computers, laptops, and mobile devices (although not optimized for mobile devices). One obvious track of future work would be to expand the code using responsive web design to become a fully functional web app which in turn would improve the overall user interface. Other options include developing a native application for either Android or iOS using the appropriate software development kit. Though costs to develop the game for the open web browser were negligible and using development kits can come at a cost, future service members are more likely to access the Internet through mobile devices whose app development process require these kits.

Though the ground warfare component of WARLOC was achieved, we were unable to completely recreate the game in the browser. An opportunity for future work would be to further develop the web-based version and add the functionality of the air planning and air execution phases of the game. The C2, intelligence, and protection joint functions were not included in the game design in order to decrease complexity and reduce gameplay time. However, the web-based version's ability to automate tasks can assist in reducing this complexity and gameplay time. It would be worth exploring methods to integrate these functions into the game without increasing either of these two factors. The game does currently contain the ability to replicate the fog of war so it may be relatively easy to integrate a simple game mechanic to familiarize players with the joint intelligence function. Expanding the fog of war also supports the expansion of the SOF component representation. The fog of war feature would enable the players ability to maneuver SOF counters throughout the enemy's territory in secret. Finally, North Korea's push into South Korea is only one of many possible scenarios. Creating additional scenarios executable within WARLOC would be another welcomed advancement of the game.

As it stands now, WARLOC is designed for two players to play the game on a single device. Adding a network capability would allow distributed players to play over the Internet. Creating this capability within WARLOC would be a great opportunity for anyone interested in exploring the usefulness of open source libraries like Node.js [71].

It is not uncommon for wargamers to play a multiplayer wargame in solitary mode. However, an obvious improvement to WARLOC would be to develop some level of artificial intelligence and create a single player mode. This could be accomplished using simple scripting techniques or by utilizing a game search technique such as minimax or alpha-beta pruning [72].

## 5.5 Limitations of this Study

The work discussed in this thesis is exploratory in nature. WARLOC is intended to familiarize its players with joint theory and functions. No user tests were conducted to assess the validity of this intention. If WARLOC could be fully developed, an experiment could be created to determine both usability and the game's effectiveness in achieving this goal.

## 5.6 Summary

The U.S. military, once again, finds itself at the crossroad of an uncertain future. Faced with an unclear operating environment and a population of experienced combat vets, the senior military leaders must determine the training direction of the armed services. In a fiscally-constrained environment, web-based learning games can assist in rapidly bridging the gap between the familiar and unfamiliar at a lower cost than typically large simulations. Further research and development of products like WARLOC could lead to improved readiness with substantial savings.

# APPENDIX A:
# Unit Task Organization

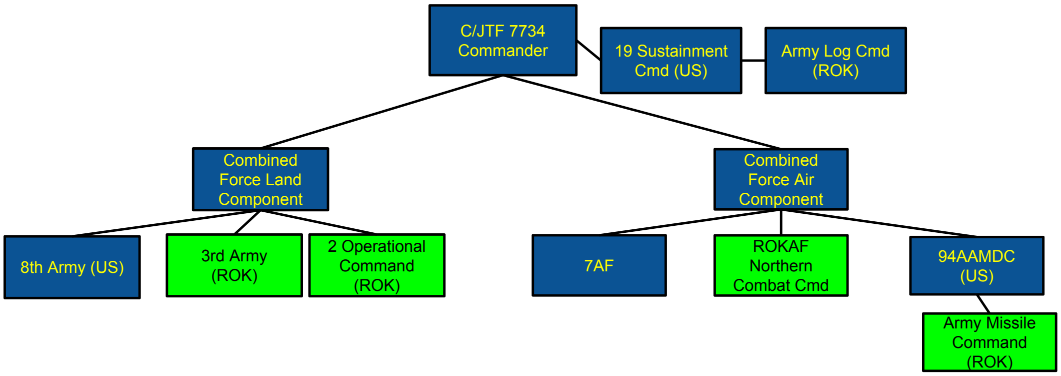

C/JTF 7734 Commander
19 Sustainment Cmd (US)
Army Log Cmd (ROK)
Combined Force Land Component
Combined Force Air Component
8th Army (US)
3rd Army (ROK)
2 Operational Command (ROK)
7AF
ROKAF Northern Combat Cmd
94AAMDC (US)
Army Missile Command (ROK)

Figure A.1: Combined Forces Command, Korea (CFC) Task Organization

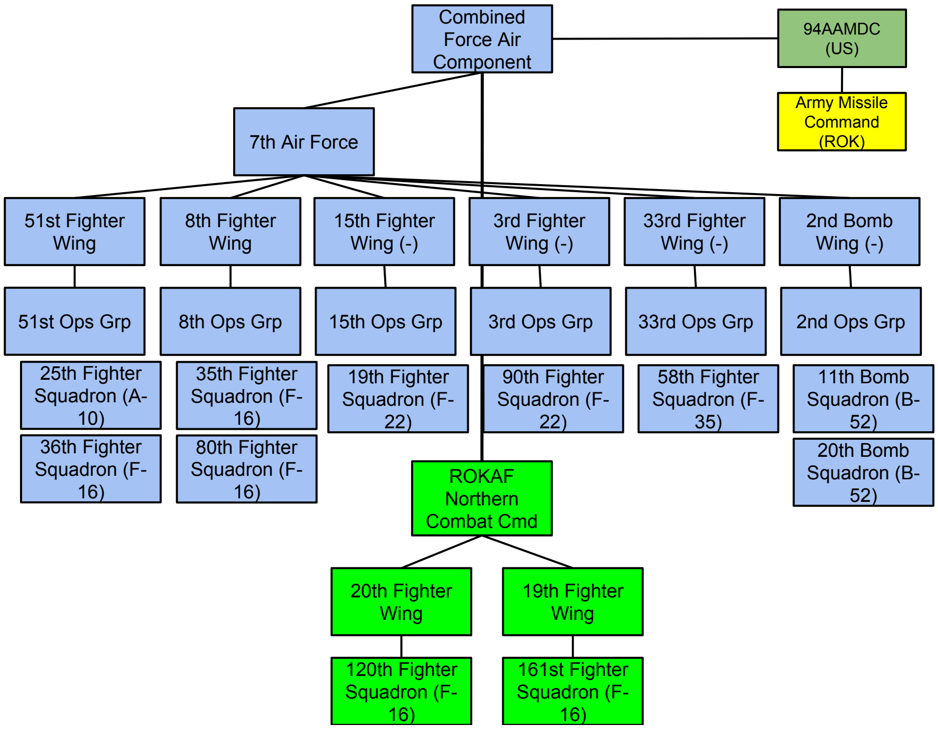

Combined Force Air Component
94AAMDC (US)
Army Missile Command (ROK)
7th Air Force
51st Fighter Wing
8th Fighter Wing
15th Fighter Wing (-)
3rd Fighter Wing (-)
33rd Fighter Wing (-)
2nd Bomb Wing (-)
51st Ops Grp
8th Ops Grp
15th Ops Grp
3rd Ops Grp
33rd Ops Grp
2nd Ops Grp
25th Fighter Squadron (A-10)
35th Fighter Squadron (F-16)
19th Fighter Squadron (F-22)
90th Fighter Squadron (F-22)
58th Fighter Squadron (F-35)
11th Bomb Squadron (B-52)
36th Fighter Squadron (F-16)
80th Bomb Squadron (F-16)
20th Bomb Squadron (B-52)
ROKAF Northern Combat Cmd
20th Fighter Wing
19th Fighter Wing
120th Fighter Squadron (F-16)
161st Fighter Squadron (F-16)

Figure A.2: Combined Fore Air Component Command (CFACC) Task Organization

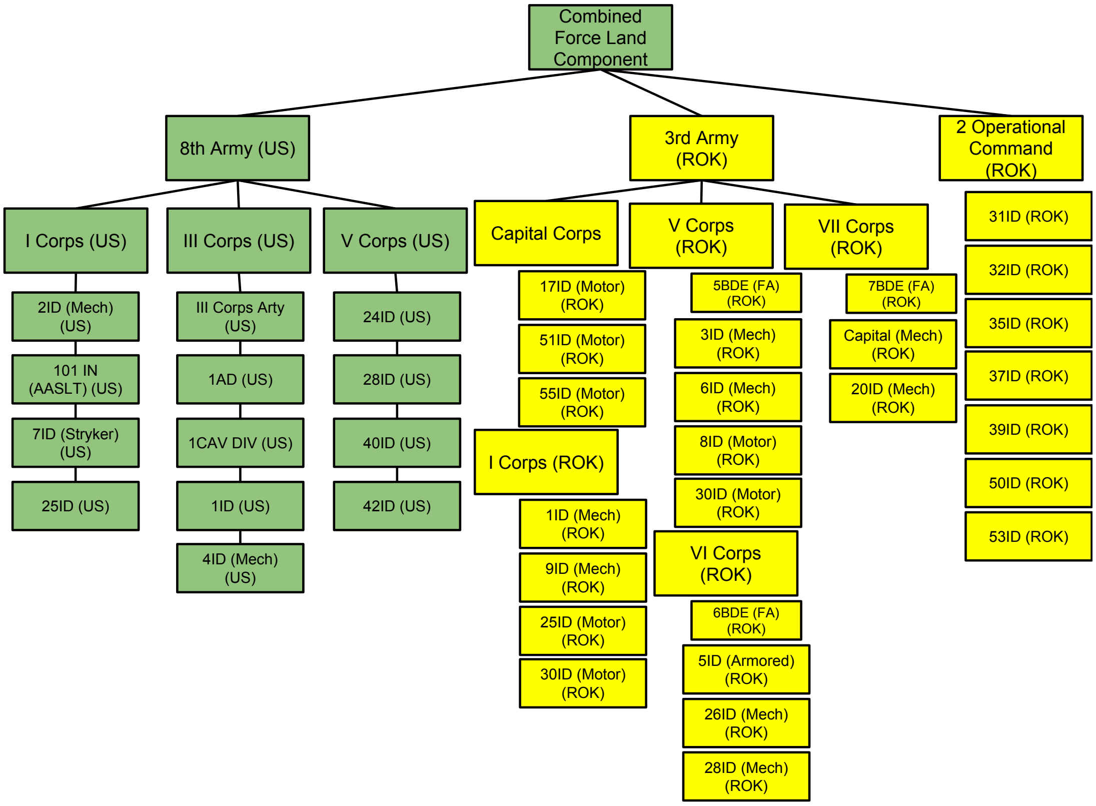


Figure A.3: Combined Forces Land Component Command (CFLCC) Task Organization

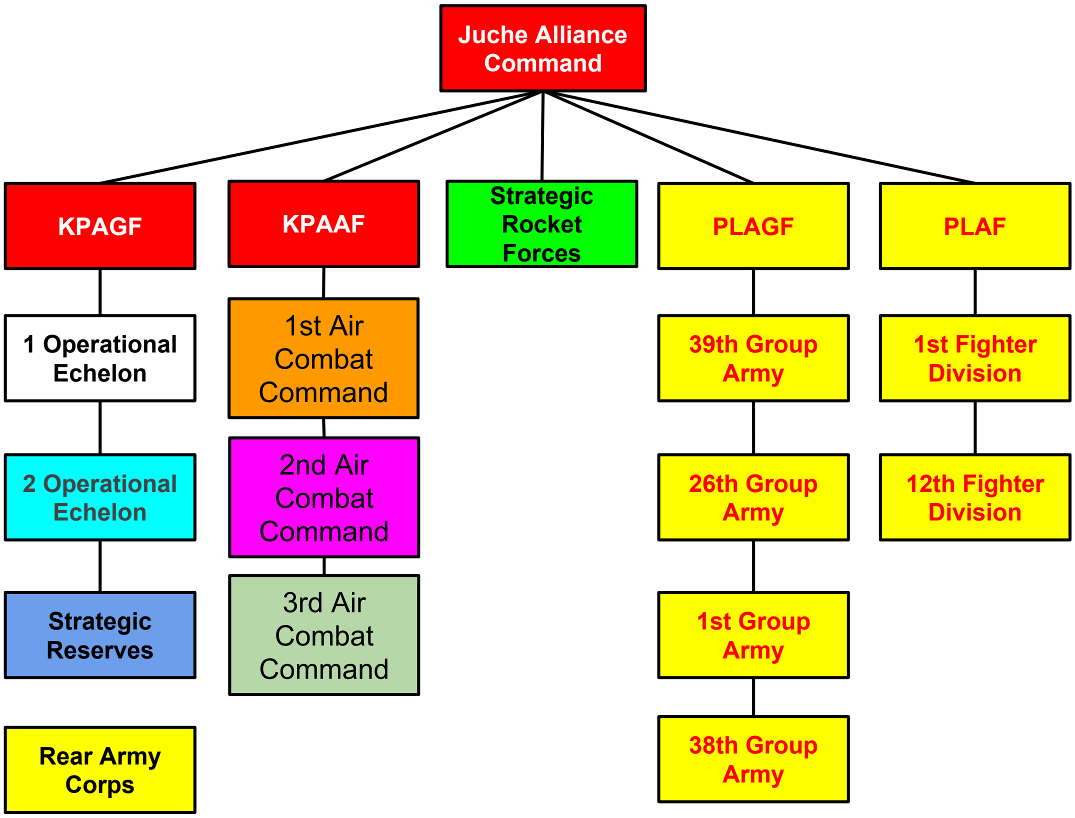


Figure A.4: Juche Alliance Task Organization

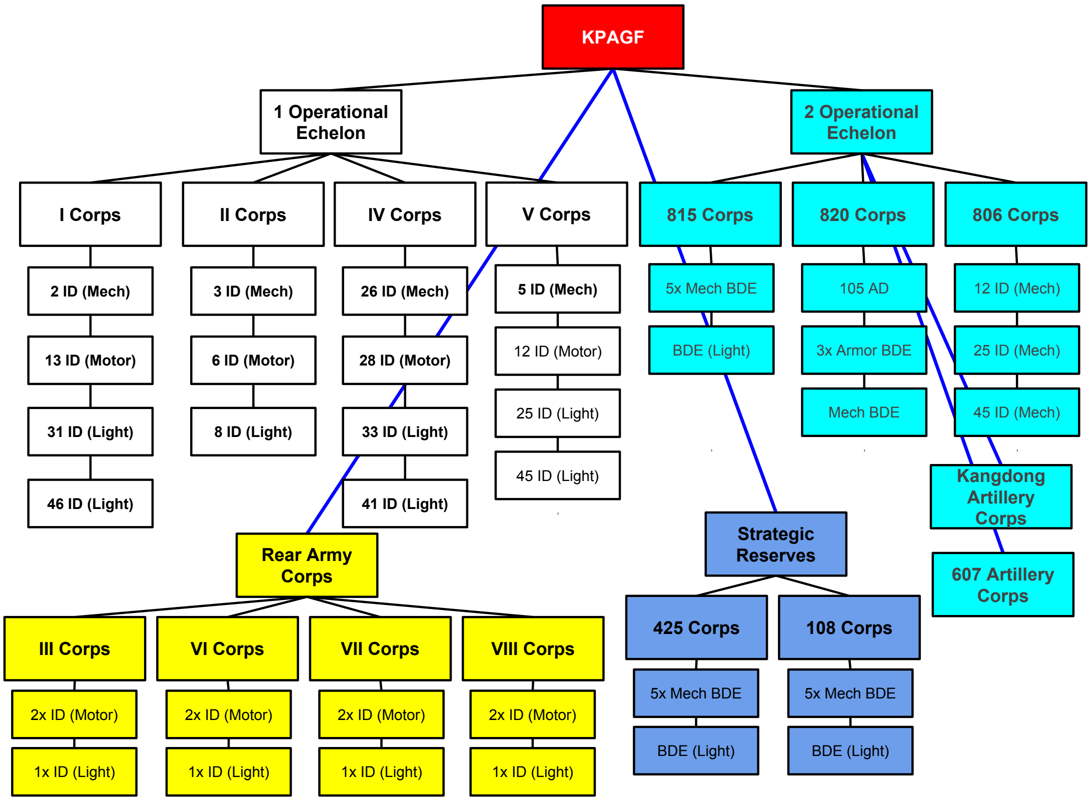


Figure A.5: Korean People's Ground Force (KPAGF) Task Organization

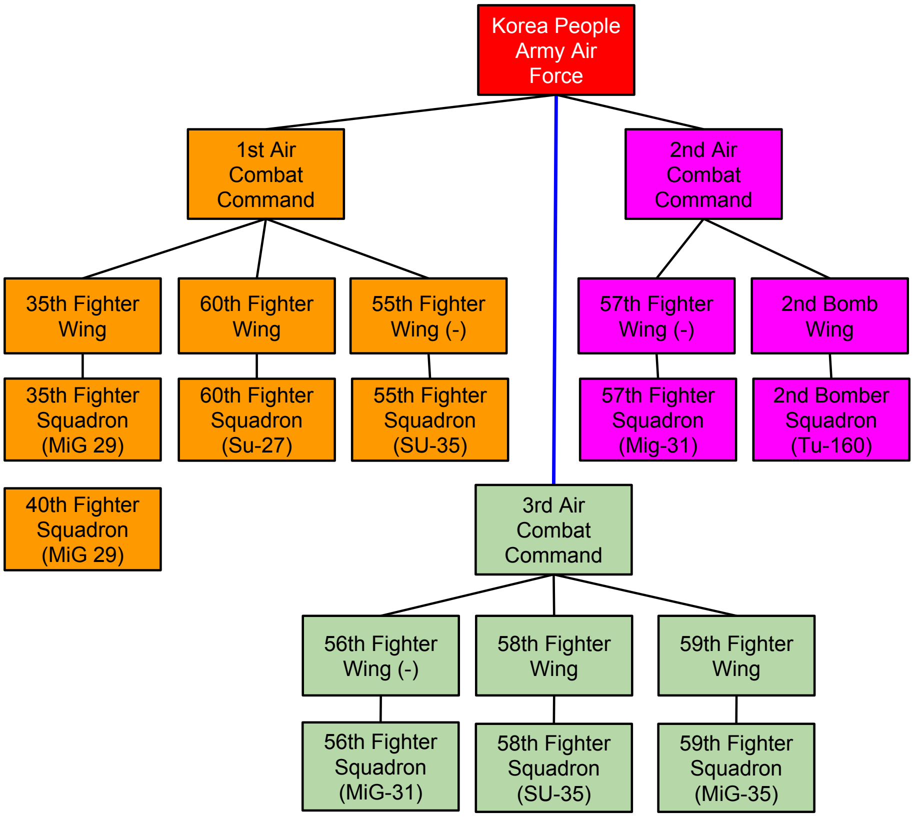

Korea People Army Air Force
1st Air Combat Command
2nd Air Combat Command
35th Fighter Wing
60th Fighter Wing
55th Fighter Wing (-)
57th Fighter Wing (-)
2nd Bomb Wing
35th Fighter Squadron (MiG 29)
60th Fighter Squadron (Su-27)
55th Fighter Squadron (SU-35)
57th Fighter Squadron (Mig-31)
2nd Bomber Squadron (Tu-160)
40th Fighter Squadron (MiG 29)
3rd Air Combat Command
56th Fighter Wing (-)
58th Fighter Wing
59th Fighter Wing
56th Fighter Squadron (MiG-31)
58th Fighter Squadron (SU-35)
59th Fighter Squadron (MiG-35)

Figure A.6: Korean People's Army Air Force(KPAF) Task Organization

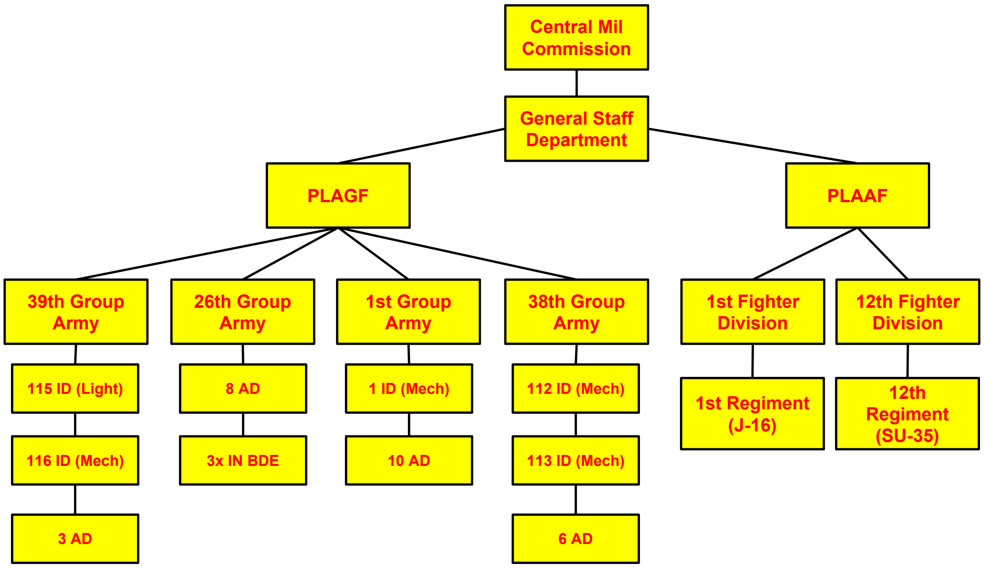
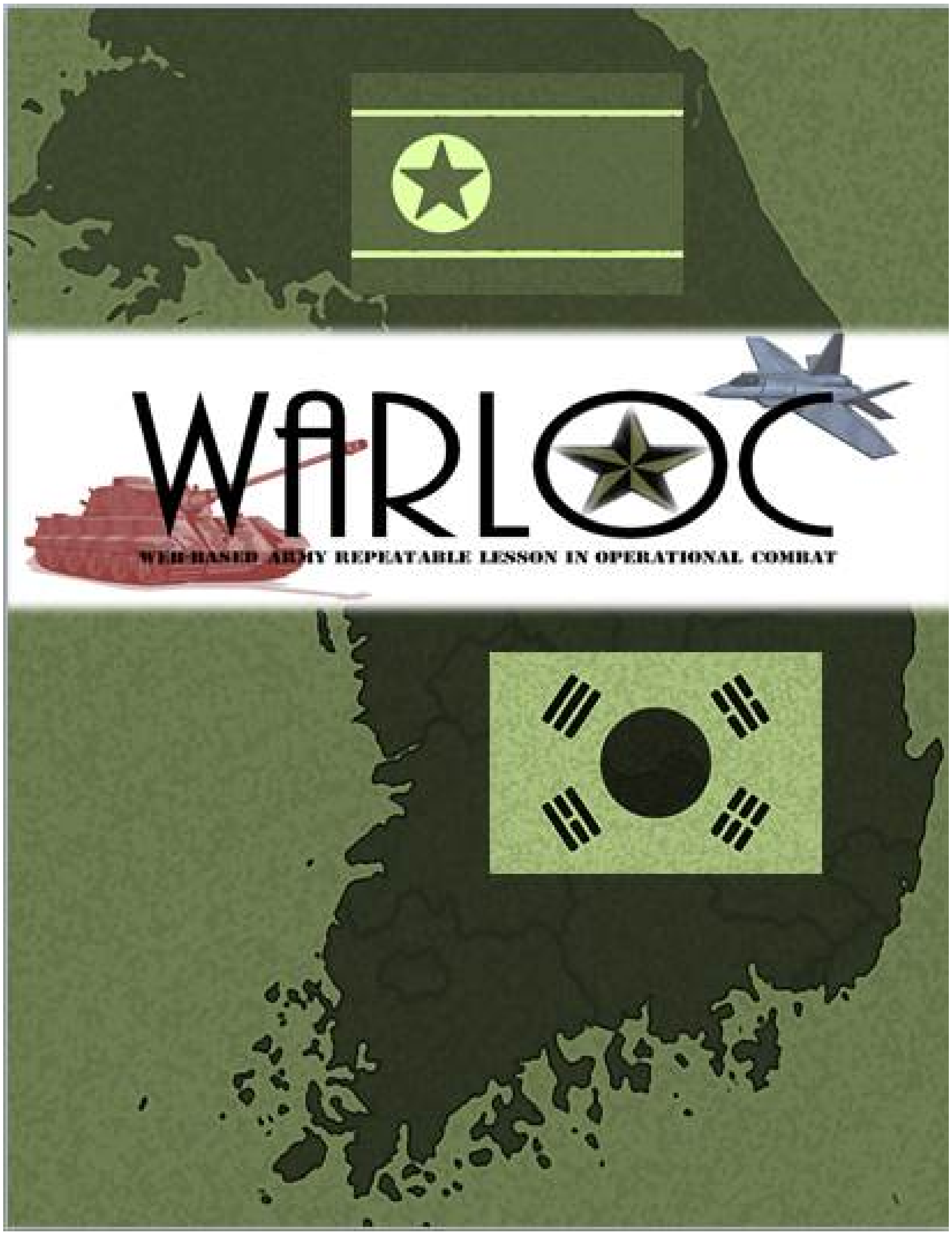


Figure A.7: People's Republic of China (PRC) Task Organization

# APPENDIX B:
# WARLOC Rule Book

This appendix includes the WARLOC's rule book in its entirety. This rule book is in a draft form and should be treated as such. Players can successfully play the WARLOC Vassal Module accompanied by these rules. The rules offer insight to anyone playing WARLOC over the web.


WARLOC
WEB-BASED ARMY REPEATABLE LESSON IN OPERATIONAL COMBAT

Version 1.1

# Table of Contents

- 4.1 First Round Modifications

5.0 Joint Fires

- 5.1 Joint Fires Targeting
  - 5.1.1 Indirect Fire Artillery Systems
  - 5.1.2 Theater Ballistic Missiles (TBM)
  - 5.1.3 Air Defense Artillery (ADA)

6.0 Air Power

- 6.1 Air Power Sequence of Play
  - 6.1.1 Air Planning Sub-Phase
  - 6.1.2 Air Operations - Force Execution Sub-Phase
- 6.2 Air Operations Description and Combat Resolution
  - 6.2.1 Air Superiority
    - 6.2.1.1 Offensive Counterair (OCA)
  - 6.2.2 Global Precision Attack
    - 6.2.2.1 Counterland Operations

7.0 Maneuver

- 7.1 Stacking
- 7.2 Zones of Control (ZOC)
  - 7.2.1 ZOC Effects
- 7.3 Terrain
  - 7.3.1 Installations
    - 7.3.1.1 Installation Occupation
    - 7.3.1.2 Installation Disruption
  - 7.3.2 Fortifications
    - 7.3.2.1 Barrier Effects
- 7.4 Movement Types
  - 7.4.1 Ground Maneuver
  - 7.4.2 Air Movement

8.0 Combat

- 8.1 Combat Eligibility
- 8.2 Combat Procedure
  - 8.2.1 Attack Declaration

# 1.0 Introduction

WARLOC (Web-based Army Repeatable Lesson in Operational Combat) is a web-based wargame in which players perform the role of a Combined Joint Task Force (C/JTF) Commander conducting major operations in a traditional war against a near-peer opponent.

## 1.1 Learning Objectives

Over a decade of persistent conflict in support of the Global War On Terror has led to a junior officer corps that is highly trained in the tactical level of war. WARLOC is an educational game, designed to familiarize junior field grade Army officers with the basics of Major Combat Operations (MCO) at the operational level. It focuses on four learning objectives:

1. Familiarization with Major Combat Operations (MCO) at the operational-level
2. Familiarization with Joint and component-level task organization
3. General understanding of the Joint Functions of Intelligence, Fires, Movement/ Maneuver, and Sustainment.
4. Introductory experience with the air power in a joint task force.

## 1.2 Scenario

North Korean leader, Kim Jung-Un's 2012 promise of becoming a "great and prosperous nation" has come to fruition through a series of aggressive economic reform and regional intimidation. Now, in the year 2022, North Korea stands at a level of economic and military prosperity that is nearly equal to its regional neighbors.

After months of provocations from both sides of the Military Demarcation Line, the United Nations Security Council is at a deadlock. Both China and Russia are staunchly against any UN or Republic of Korea (ROK) - United States offensive operations into North Korea. Seizing this period of indecision, North Korea launches an attack into South Korea with the intent to conduct a swift war while threatening nuclear retaliation if ROK-US forces enter North Korea. The People's Republic of China, while against a North Korean invasion into South Korea, is also intent on preventing a nuclear engagement. In response the PRC is supporting the defense of North Korea by augmenting North Korea's defensive ground and air forces. In addition they are providing a naval presence in the Korea Bay off of

North Korea's western coast. Meanwhile, Russia conducts naval "exercises" in the Sea of Japan directly off the eastern coast of North Korea effectively preventing ROK-US naval operations on either coast.

Players can play the role of either the Combined Forces Command (CFC) or the Juche Alliance (JA) of North Korea and PRC ground and air forces.

## 1.3 Player Roles

The players represent the two Joint Force Commanders (JFC) controlling both land and air forces at the Operational Level of War. One player will assume the role as commander of CFC and the other as the commander of the JA.

## 1.4 Victory Conditions

The players represent the two Joint Force Commanders (JFC) controlling both land and air forces at the Operational Level of War. One player will assume the role as commander of CFC and the other as the commander of the JA.

### 1.4.1 CFC Victory Conditions

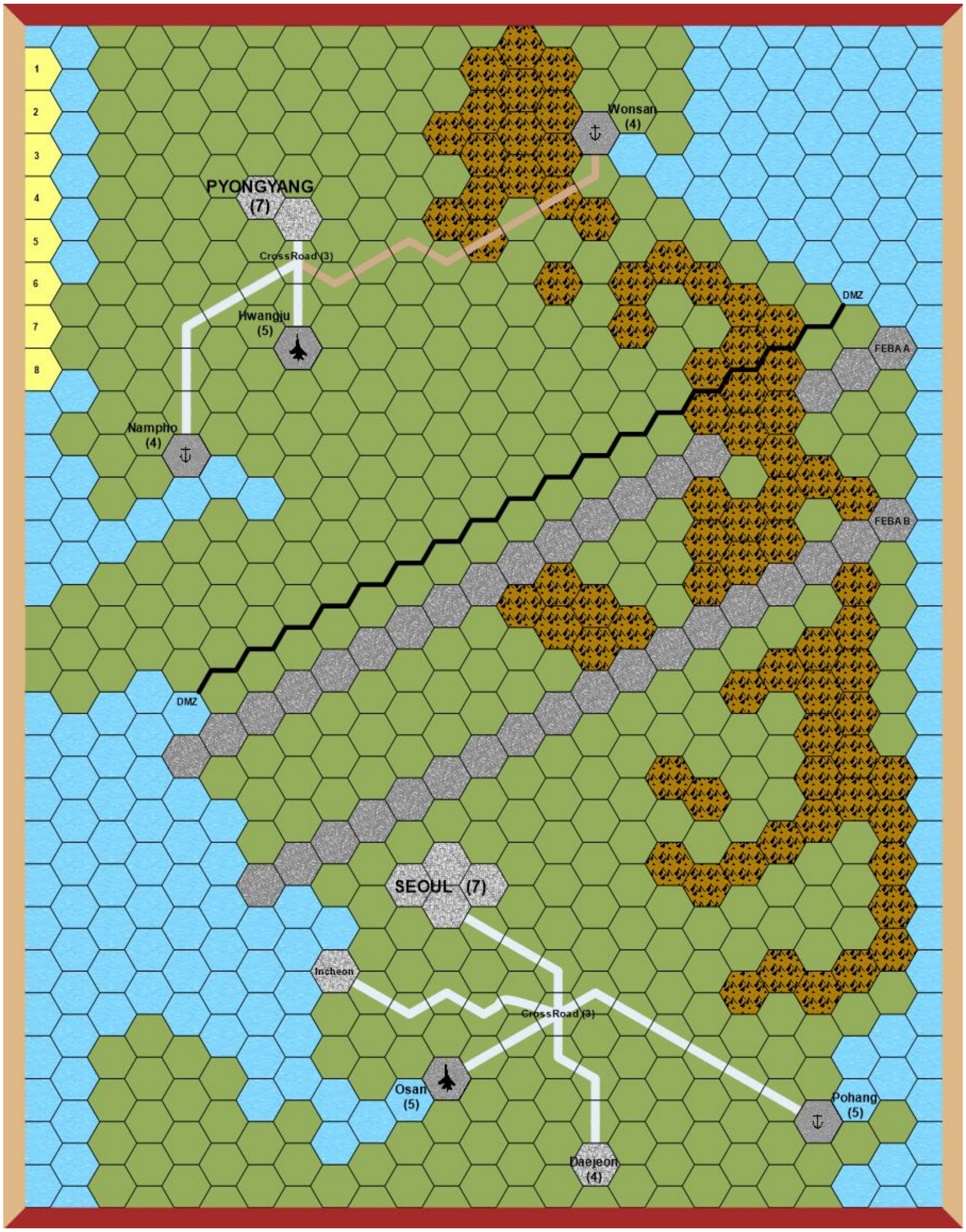
The CFC Commander is successful if they prevent JA forces from entering a Seoul hex by the end of round 8.

### 1.4.2 Juche Alliance Victory Conditions

The JA Commander is successful if any of their ground forces enter a Seoul hex by the end of round 8.

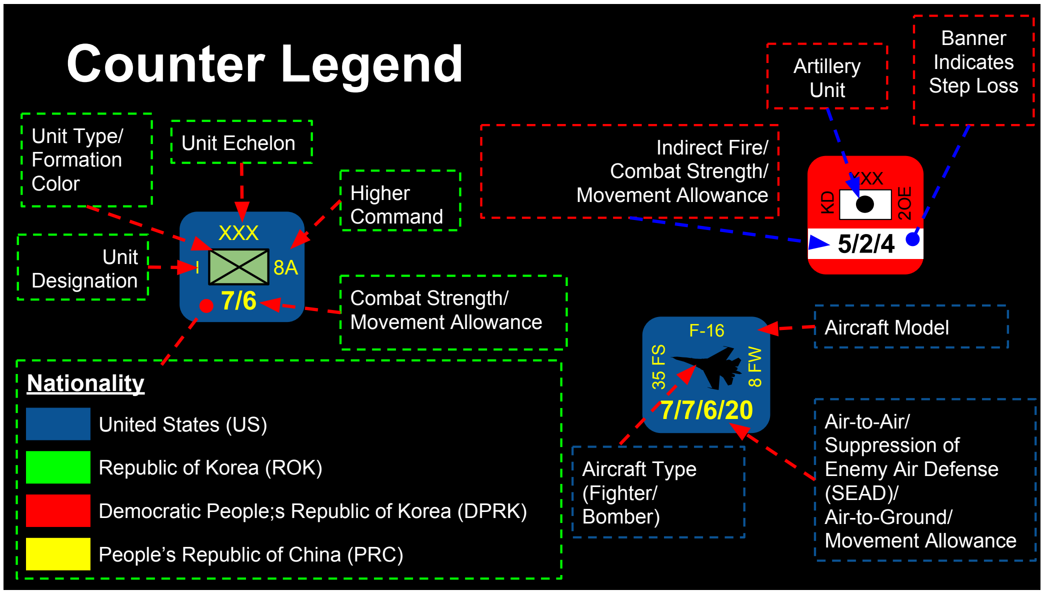
# 2.0 Components

## 2.1 Map



WARLOC Game Map

## 2.2 Counters


Counter Legend
Unit Type/ Formation Color
Unit Echelon
Higher Command
Unit Designation
Combat Strength/ Movement Allowance
XXX
I
8A
7/6
Indirect Fire/ Combat Strength/ Movement Allowance
Artillery Unit
Banner Indicates Step Loss
XXX
KD
2OE
5/2/4
Aircraft Model
F-16
35 FS
8 FW
7/7/6/20
Aircraft Type (Fighter/ Bomber)
Air-to-Air/ Suppression of Enemy Air Defense (SEAD)/ Air-to-Ground/ Movement Allowance
Nationality
United States (US)
Republic of Korea (ROK)
Democratic People;s Republic of Korea (DPRK)
People's Republic of China (PRC)

Counter Legend

## 2.3 Explanation of Universal Common Counter Values

### 2.3.1 Movement Allowance

Represents the maximum number of Movement Points (MPs) a unit can expend during each turn.

### 2.3.2 Unit Designation/Higher Command

Each individual unit (Unit Designation) belongs to a higher formation/command (Higher Command). The color of the unit symbol also distinguishes the unit's command for quick reference.

## 2.4 Explanation of Ground Specific Values

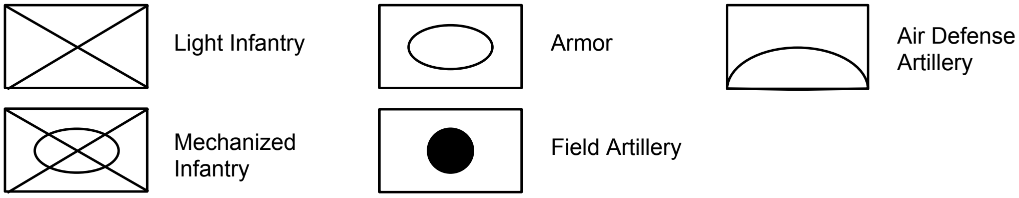
### 2.4.1 Unit Echelon

Ground Unit Sizes: Each ground unit has a unit size marked directly above its unit symbol.

XX - Division
XXX - Corps

### 2.4.2 Unit Type

Ground Units are represented by standard NATO symbols.



Unit Symbols

### 2.4.3 Combat Power

An estimated numerical value of a unit's morale, training, doctrine, cohesion, and ancillary weapons systems. This value is used to resolve all Direct Fire combat in both the attack and the defense.

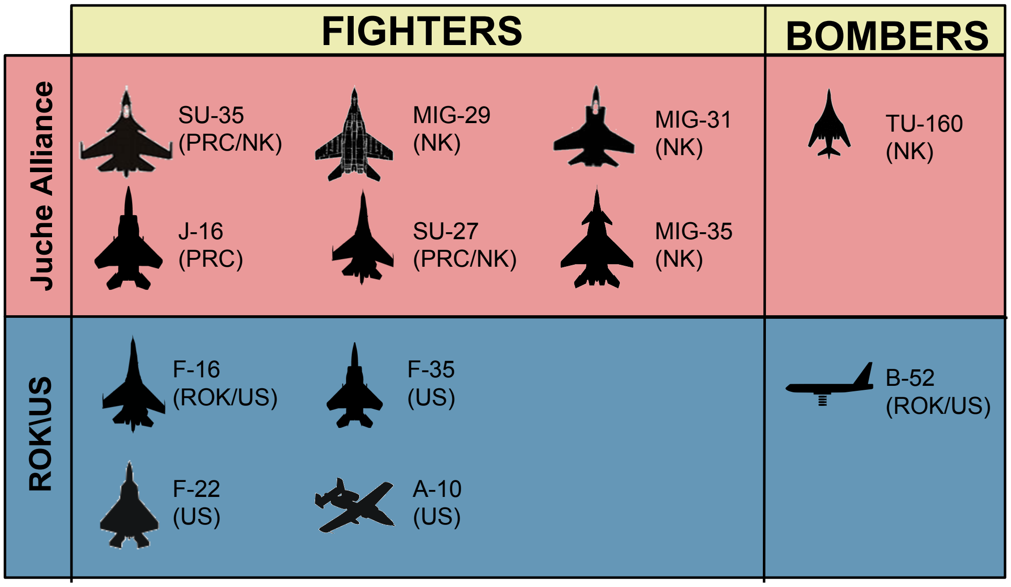
### 2.4.4 Indirect Fire

The combat power of an artillery unit conducting indirect fire attacks.

## 2.5 Explanation of Air Unit Specific Values

### 2.5.1 Aircraft Type/Model

Aircraft silhouettes represent the various aircraft models as follows:



Aircraft Models

### 2.5.2 Air-to-Air

The attacking/defending strength used in air-to-air direct fire combat (fighter vs fighter)

### 2.5.3 Suppression of Enemy Air Defense (SEAD)

Attack strength used against an Air Defense Artillery system during a SEAD mission.

### 2.5.4 Air-to-Ground

Attack strength of any air-to-ground attack, units or infrastructure, EXCEPT for attacks on air defense artillery (ADA) units (use SEAD combat power for this combat resolution).

## 2.6 Combat Resolution Table (CRT)

There are four Combat Results Tables (CRT) to choose from:

1. Direct Fire [DF] (Ground to Ground, Air to Air)
2. Indirect Fire [IDF] (Artillery, Surface-to-Surface Missile)
3. Surface to Air [S2A] (Air Defense Artillery Fire on Attacking Aircraft)
4. Air to Ground [A2G] (Air Interdiction, Close Air Support, Suppression of Enemy Air Defense, Attack Operations)

### 2.7 Game Scale

The map scale is 7.5 miles per hex. Unit scale is generally corps, although some independent weapon systems are used to represent ADA and TBM sites. Each game turn represents roughly one day.

### 2.8 Hex Control

Units control the hex in which they reside as well as all hexes in their Zone of Control (ZOC). Hexes overlapped by ZOCs of two opposing units is considered "contested." However, in general, at the start of the game all hexes south of the Demilitarized Zone (DMZ) are controlled by the ROK/US Combined Forces Command (CFC) while all hexes north of the DMZ are controlled by the Juche Alliance.

# 3.0 Game Setup

### 3.1 Game Map Setup

In an effort to simulate operational surprise and the delayed arrival of PRC forces, some unit counters are not immediately available for use and should be placed as indicated.

1. The following forces are not available until the players turn in round 2 and should be stacked on the '2' found in the round tracker on the game board. (Note: Place 3rd Corps on top as it is possible for this unit's arrival to be delayed.)
   (a) 3rd Corps 8th Army
   (b) ROK CFTF (x2)
   (c) US 58th Fighter Squadron, 33rd Fighter Wing
   (d) US 19th Fighter Squadron, 15th Fighter Wing
   (e) US 90th Fighter Squadron, 3rd Fighter Wing
2. The following forces are not available until the players turn in round 3 and can be stacked on the '3' found in the round tracker on the game board.
   (a) US 11th Bomber Squadron, 2nd Bomber Wing
   (b) US 20th Bomber Squadron, 2nd Bomber Wing
3. The following forces are not available until the JA Commander's turn in round 4 and can be stacked on the '4' found in the round tracker on the game board.
   (a) China 1st Fighter Detachment, 1st Fighter Regiment

    (b) China 12th Fighter Detachment, 12th Fighter Regiment

4. The following forces are not available until the JA Commander's turn in round 6 and can be stacked on the '6' found in the round tracker on the game board.
    (a) China 1st PLAGF
    (b) China 26th PLAGF
    (c) China 38th PLAGF
    (d) China 39th PLAGF
5. Air Units for both Commanders are conducting CounterAir Patrols (CAP) so are not available for the first Air Planning phase and are located at the following.
    (a) KPA 1st ACC, 60th Fighter Wing - Located 1 hex outside of any Pyongyang hex
    (b) KPA 1st ACC, 35th Fighter Wing - Located 1 hex outside of Hwangju
    (c) US 35th Fighter Squadron, 8th Fighter Wing - Located 1 hex outside of Seoul

Players may freely place all other ground units with the following exceptions:

1. No JA forces South of the DMZ
2. NO CFC forces North of FEBA A
3. ROK Capital Corps (CAP 3A) must be in one of Seoul's 4 hexes.

### 3.2 Air Tasking Order (ATO) Cards

All air counters not currently placed on the turn tracker on the game map are placed within the "Sorties Available" row on the player's ATO card.

## 4.0 Standard Sequence of Play

WARLOC is played in 8 rounds consisting of 2 turns, 1 for each player.
Each turn is divided into specific, ordered Phases.
The Sequence of Play consists of the following Phases:

1. Apply any effects on remaining friendly forces from enemy's attack the previous turn
2. Joint Fires Operations
    (a) Air Operations - Force Execution
        i. Air Movement

    ii. Air Combat (refer to 6.2 Air Operations and Combat Resolution)
        A. Air-to-Air Combat [DF]
        B. Deliberate or Dynamic SEAD [A2G]
        C. Surface-to-Air Combat (ADA vs Air Threat) [S2A]
        D. Air-to-Ground combat (Attack Ops, Air Interdiction, Close Air Support) [A2G]
  (b) Ground-based Indirect Fires

3. Ground Operations
   (a) Ground Movement
   (b) Ground Combat
4. Air Planning - Air Sorties Allotment for following turn
5. Housekeeping (remove disruption counters and adjust reinforcement arrival if applicable)

## 4.1 First Round Modifications

Once the game is setup both players will have their ATO card's "Sorties Available" section containing taskable air unit counters. During the first round and first round only, both players SIMULTANEOUSLY conduct their Air Planning sub-phase. Defensive CounterAir units are already in pllace on the map.

Note: Typically players will conduct their Air Planning sub-phase sequentially during their game turn as described in the Air Planning Sub-Phase section, however the first game turn is unique and the sequence of events is as follows:

1. Both players conduct the Air Planning for the first turn during the game setup portion. The players then use this plan to execute their Joint Fires portion of their 1st round game turn.
2. At the end of each player's turn in round 1 they will conduct their Air Planning Phase for their next turn in round 2 and the Standard Sequence of Play will naturally occur for the remainder of the game.

# 5.0 Joint Fires

**Joint Fires.** *Fires delivered during the employment of forces from two or more components in coordinated action to produce desired effects in support of a common objective.*
**Joint Fires Support** *Joint fires that assist air, land, maritime, and SOF to move, maneuver, and control territory, populations, airspace, and key waters in support of the joint force commander's (JFC's) scheme of maneuver.*

-JP 3-09(Joint Fire Support)

*Design Note: One of the game's learning objectives is for players to create a single battle plan that synchronizes both maneuver and fires. Additionally, they should learn how to integrate air operations and ground operations into their fires plan*

In WARLOC, Joint Fire Support is conducted by forces from both the land component and air component. This section discusses ground indirect fire systems and missile systems. Joint Fires combat is explained in the Combat section.

## 5.1 Joint Fires Targeting

Each joint fires system has specific targets that can be engaged within its given range.

### 5.1.1 Indirect Fire Artillery Systems

Ground artillery units have both direct fire (indicated by the unit's combat strength) and indirect fire capabilities. Direct fire combat is covered in the Combat section.

**Artillery Unit Specifications**

1. **Range:** 10 hexes.
2. **Targets:** Any ground enemy unit, Artillery CANNOT engage airborne targets using its indirect fire value. It can defend itself against aircraft using its direct fire combat power value.
3. **Associated CRT:** Indirect Fires

### 5.1.2 Theater Ballistic Missiles (TBM)

Ground artillery units have both direct fire (indicated by the unit's combat strength) and indirect fire capabilities. Direct fire combat is covered in the Combat section.

**SRF Unit Specifications**

1. **Range:** 25 hexes.
2. **Targets:** Infrastructure (C2 Nodes, Air Bases, Logstic Hubs, Ports) and crossroads.
3. **Associated CRT:** 2 possible cases

   (a) **Targeted infrastructure is within a friendly ADA units ZOC and ADA is not disabled (Protected).**

      i. Roles: Aircraft = Attacker, ADA = Defender
      ii. Differential: Defender (Indirect Fire Value + Infrastructure Defense Value) - Attacker Indirect Fire Value
      iii. CRT: S2A

         A. AE: No Effect on targeted Infrastructure (ADA successfully shot down TBM)
         B. NE, AR: Targeted infrastructure is disabled (ADA failed to shoot down TBM). Apply Interdiction effects as described in 9.0 Command & Supply Effects.

   (b) **Targeted infrastructure is NOT within a friendly ADA units ZOC or protecting ADA system is disabled (Unprotected).**

      i. Roles: Aircraft = Attacker, Target = Defender
      ii. Differential: Attacker A2G Value - Defender infrastructure defense value
      iii. CRT: A2G

         A. DE, Disrpt: Targeted infrastructure is disabled. Apply Interdiction Affects as described in 9.0 Command & Supply Effects.
         B. NE, AE: TBM missed its target, No Effect.

### 5.1.3 Air Defense Artillery (ADA)

*Design Note:*

1. ***Point vs. Area Defense.*** *n order to simplify gameplay, we have focused on the point defense portion of air defense and not on the area defense portion. That is why the Zone of Control for an ADA unit is limited to the six hexes surrounding an ADA unit.*
2. ***ADA detection range.*** *We are portraying a short-medium detection range for ADA systems. We are also depicting that both sides have offensive airborne C2 assets that are coordinating the offense counterair operation. This gives an aircraft flying a SEAD mission a detection range advantage over the ADA system.*

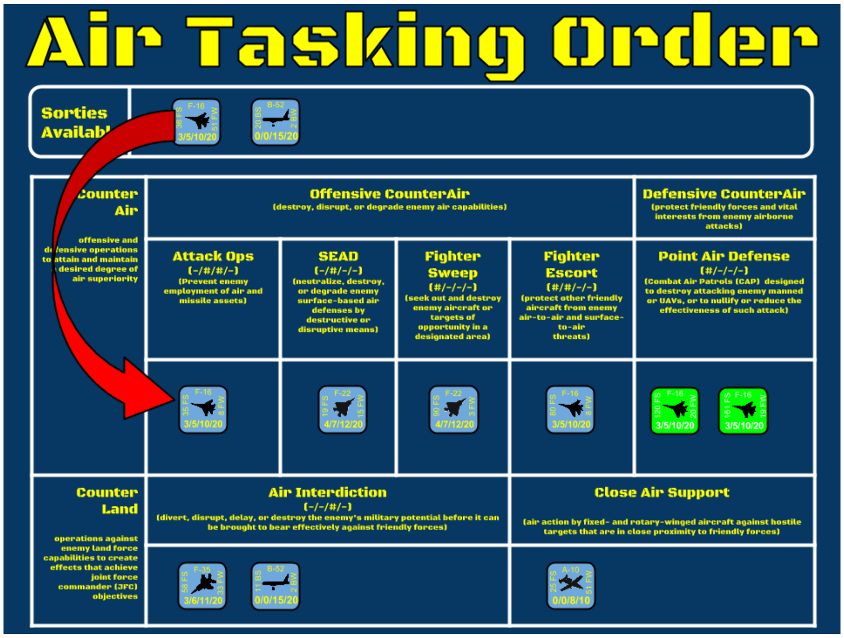
1. **Range** ADA unit's ZOC
2. **Targets** Aircraft and TBMs.
3. **Associated CRT**
   (a) Aircraft and TBMs: Use Indirect Fire Value (1st value on counter) and S2A CRT
   (b) Ground Units: Use Direct Fire Value (2nd value on counter) and DF CRT
4. NOTE: If an ADA unit is disabled it is rendered ineffective unable to defend against air threats. In this case, disregard the systems protective umbrella (Air ZOC) and do not add the ADA indirect fire combat value to the defense value of the infrastructure for combat resolution purposes. Disabled ADA systems can defend themselves from attack (Direct Fire or SEAD) but use the ADA system's DF combat power to resolve this type of combat.

# 6.0 Air Power

*Airpower is the ability to project military power or influence through the control and exploitation of air, space, and cyberspace to achieve strategic, operational, or tactical objectives.*

*-Air Force Doctrine Document (AFDD) 1 (Air Force Basic Doctrine Organization, and Command)*

The ATO Card is used to plan and execute air operations. It is divided into two sections, sorties available and air operations. The ATO is used to plan all air operations to be conducted during the air operation execution phase.



Air Tasking Order

Sorties Available

Counter Air

offensive and defensive operations to attain and maintain a desired degree of air superiority

Offensive CounterAir (destroy, disrupt, or degrade enemy air capabilities)

Defensive CounterAir (protect friendly forces and vital interests from enemy airborne attacks)

Attack Ops (-/#/#/-) (Prevent enemy employment of air and missile assets)

SEAD (-/#/-/-) (neutralize, destroy, or degrade enemy surface-based air defenses by destructive or disruptive means)

Fighter Sweep (#/-/-/-) (seek out and destroy enemy aircraft or targets of opportunity in a designated area)

Fighter Escort (#/#/-/-) (protect other friendly aircraft from enemy air-to-air and surface-to-air threats)

Point Air Defense (#/-/-/-) (Combat Air Patrols (CAP) designed to destroy attacking enemy manned or UAVs, or to nullify or reduce the effectiveness of such attack)

Counter Land

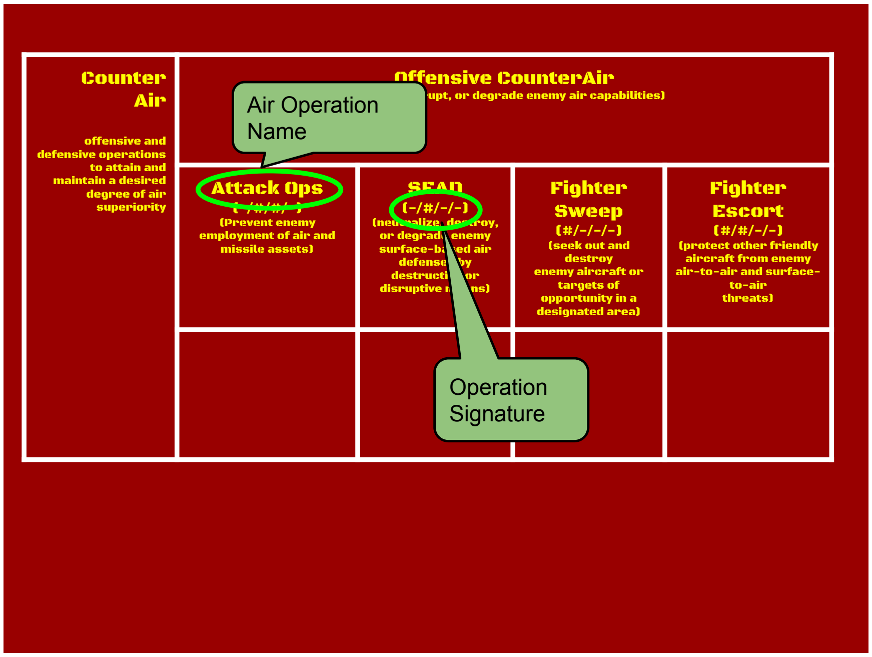
operations against enemy land force capabilities to create effects that achieve joint force commander (JFC) objectives

Air Interdiction (-/-/#/-) (divert, disrupt, delay, or destroy the enemy's military potential before it can be brought to bear effectively against friendly forces)

Close Air Support (air action by fixed- and rotary-winged aircraft against hostile targets that are in close proximity to friendly forces)

F-16 36 FS 51 FW 3/5/10/20
B-52 20 BS 2 BW 0/0/15/20
F-16 35 FS 8 FW 3/5/10/20
F-22 19 FS 15 FW 4/7/12/20
F-22 90 FS 3 FW 4/7/12/20
F-16 80 FS 8 FW 3/5/10/20
F-16 120 FS 20 FW 3/5/10/20
F-16 161 FS 19 FW 3/5/10/20
F-35 58 FS 33 FW 3/6/11/20
B-52 11 BS 2 BW 0/0/15/20
A-10 25 FS 51 FW 0/0/8/10

Air Tasking Order Player Card with Air Unit Counters



Counter Air

offensive and defensive operations to attain and maintain a desired degree of air superiority

Offensive CounterAir (destroy, disrupt, or degrade enemy air capabilities)

Attack Ops (-/#/#/-) (Prevent enemy employment of air and missile assets)

SEAD (-/#/-/-) (neutralize, destroy, or degrade enemy surface-based air defenses by destructive or disruptive means)

Fighter Sweep (#/-/-/-) (seek out and destroy enemy aircraft or targets of opportunity in a designated area)

Fighter Escort (#/#/-/-) (protect other friendly aircraft from enemy air-to-air and surface-to-air threats)

Air Tasking Order Card elements

## 6.1 Air Power Sequence of Play

Air Power is divided into two separate sub-phases:

1. **Air Planning** - Sorties Allotment
2. **Air Operations** - Force Execution

### 6.1.1 Air Planning Sub-Phase

> *Design Note: The Air Planning Sub-Phase represents a combination of several stages in the Air Tasking Cycle:*
>
> 1. ***Objectives, Effects, and Guidance Stage.*** *It replicates the JFC apportionment which is the total expected effort that should be devoted to the various airpower operations for a given period of time (in this case, a single game turn).*
> 2. ***Weaponeering and Allocation stage.*** *The apportionment is turned into an allocation of forces.*

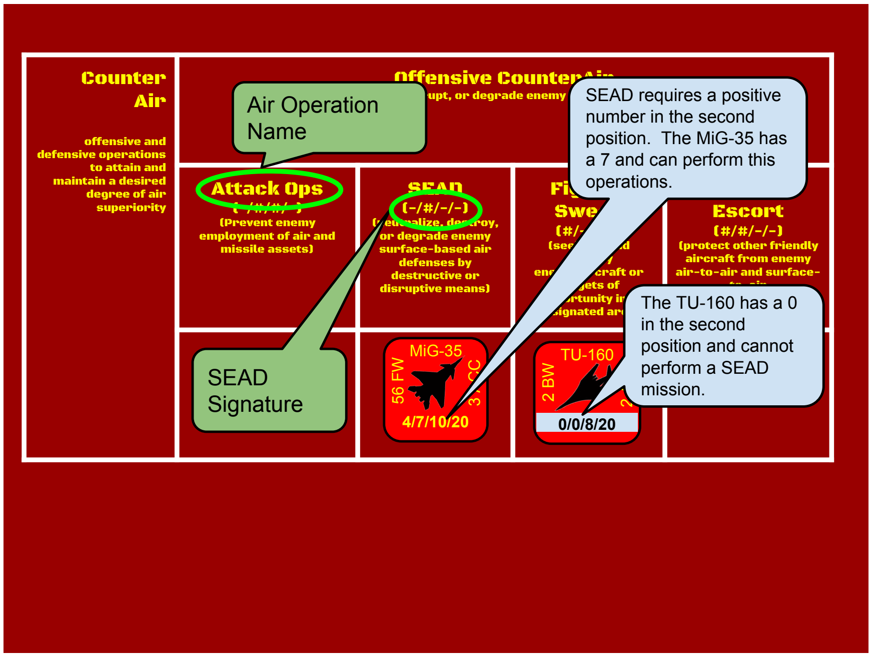
**Note:** The ATO created in this phase is used during the Force Execution Phase **on the following game turn** (with the exception of Game Turn 1, where the planned ATO is immediately executed on that same turn).

Each air unit counter has a series of values underneath the aircraft silhouette. These values represent the unit's combat strength/movement allowance as described in section 2.5 (Explanation of Air Unit Values). In order for a unit to perform an air operation it must have a value greater than 0 in the matching position listed in the air operation signature (see figure below).

Players assign air units to specific air operations (an explanation of the individual air operations is described in subsequent sections) by dragging them from the "Sorties Available" row into the appropriate Air Operation box. The air unit's values must match the Air Operation Signature in order to be assigned to the mission.

### 6.1.2 Air Operations - Force Execution Sub-Phase

Players start with the ATO card configured during the Air Planning Sub-phase. Players move the counters from the individual Air Operation boxes to their desired target locations (all planned air sorties are moved to the game map at this point). Each air unit's mobility range is determined by it's current movement points using the player's air base as the units


Counter Air
offensive and defensive operations to attain and maintain a desired degree of air superiority
Offensive CounterAir
upt, or degrade enemy
Attack Ops
(-/#/#/-)
(Prevent enemy employment of air and missile assets)
SEAD
(-/#/-/-)
(neutralize, destroy, or degrade enemy surface-based air defenses by destructive or disruptive means)
Fi
Swe
(#/-
(see
ene craft or
gets of
rtunity in
signated ar
Escort
(#/#/-/-)
(protect other friendly aircraft from enemy air-to-air and surface-to-air
MiG-35
56 FW
4/7/10/20
TU-160
2 BW
0/0/8/20

Air Operation Name

SEAD Signature

SEAD requires a positive number in the second position. The MiG-35 has a 7 and can perform this operations.

The TU-160 has a 0 in the second position and cannot perform a SEAD mission.

Air Unit Capable of Performing SEAD

origin (Osan for the CFC Commander and Hwangju for the JA Commander). Air counters are placed either on top or next to the intended target based on required map visibility. Combat is resolved according to the Sequence of Play (Section 4.0) using the appropriate CRT.

During the Force Execution Sub-Phase, Aircraft may be re-assigned to any mission type regardless of the box they occupy in the ATO card. However, if this occurs, the air unit will incur a Penalty.

**IF PLAYERS CHANGE MISSION TYPES THEY INCUR -1 ATTACK DRM**

## 6.2 Air Operations Description and Combat Resolution

For the purpose of air operations, all combat is pair-wise (1:1). For example, if infrastructure is protected by an ADA system and an aircraft (CAP), in order for the phasing player to successfully engage the infrastructure with the bomber he/she will need to plan a SEAD mission to handle the ADA, as well as a Fighter Sweep/Escort mission to engage the CAP, allowing the bomber to freely attack the infrastructure. Each of these combat situations will be resolved independently of one another. If the bomber is sent alone, or only with

SEAD or an Escort, either the CAP or the ADA will engage the bomber (whichever is not engaged) and the bomber will never engage the infrastructure. NOTE: This is a bad day for the bomber as their DF and SEAD values are 0 meaning they are virtually defenseless without some sort of support.

### 6.2.1 Air Superiority

> ***Air superiority*** *is that degree of dominance in the air battle of one force over another that permits the conduct of operations by the former and its related land, maritime, and air forces at a given time and place without prohibitive interference by the opposing force's air and missile threats. Historically, air superiority has proven to be a prerequisite to success for an operation/campaign because it prevents enemy air and missile threats from interfering with operations of friendly air, land, maritime, space, and special operations forces, assuring freedom of action and movement*
>
> *-JP 3-01 (Countering Air and Missile Threat)*

Sub-elements of Air Superiority are Offensive Counterair (OCA) and Defensive Counterair (DCA).

Note: DCA Operations consist of both Active and Passive Measures (JP 3-01 (Countering Air and Missile Threat)). The game replicates the Active Measures in the Fires Section.

#### 6.2.1.1 Offensive Counterair (OCA)

> *OCA is offensive operations to destroy, disrupt, or neutralize enemy aircraft, missiles, launch platforms, and their supporting structures and systems both before and after launch, and as close to their source as possible. The goal of OCA operations is to prevent the launch of enemy aircraft and missiles by destroying them and their overall supporting infrastructure prior to employment. is offensive operations to destroy, disrupt, or neutralize enemy aircraft, missiles, launch platforms, and their supporting structures and systems both before and after launch, and as close to their source as possible. The goal of OCA operations is to prevent the launch of enemy aircraft and missiles by destroying them and their overall supporting infrastructure prior to employment.*
>
> *-JP 3-01 (Countering Air and Missile Threat)*

1. **Attack Operations** Include targeting enemy air and missile forces on the surface and the infrastructure and systems that contribute to their capabilities. Some Services refer to these as "strike" operations. **(Game Use: Sorties tasked with this mission are able to attack infrastructure - not much different than interdiction)**
   (a) Attack Operations Targets
      i. Air and missile unit C2 nodes/centers.
      ii. Ballistic Missiles (BM) on fixed and mobile launchers.
      iii. Airfield runways, taxiways, and underground facilities entrances.
   (b) Combat Resolution
      i. **(Protected)** If ADA is (1) present, (2) not the intended target, and (3) not disabled, the aircraft is engaged by the ADA system. If the aircraft survives the ADA encounter it will then engage its intended target as an unprotected target listed next.
         A. Roles: ADA = Attacker, Aircraft = Defender
         B. Differential: Attacker Indirect Fire Value - Defender SEAD value
         C. CRT: S2A
      ii. **(Unprotected).** If ADA is not present or is disabled the aircraft freely attacks its intended target.
         A. Differential: Aircraft A2G value - Targets (DF) combat power or infrastructure defense value
         B. CRT: Attacking Enemy Unit = DF, Attacking Enemy Infrastructure = A2G
         C. Combat Result: Aircraft = Attacker, Target = Defender
2. **SEAD.** Activity that neutralizes, destroys, or temporarily degrades surface-based enemy air defenses (AD) by destructive and/or disruptive means. **(GAME USE: Sorties planned for SEAD missions will attack ADA systems directly. This prevents the ADA from targeting other Air Sorties that may be conducting an Interdiction mission.)**
   (a) SEAD Targets
      i. ADA
   (b) Combat Resolution
      i. **Deliberate SEAD Mission**

A. Roles: Aircraft = Attacker, ADA = Defender

B. Differential: Attacker SEAD value - Defender Indirect Fire Value

C. CRT: A2G

ii. **Dynamic SEAD Mission**

A. Follow Deliberate SEAD steps but -2 DRM is applied to the dice roll

3. **Fighter Escort.** Fighter escort includes providing dedicated protection sorties by air-to-air capable fighters in support of other offensive air and air support operations over enemy territory or in a DCA role to protect aircraft such as high value airborne assets (HVAAs). Fighter escort sorties are normally flown over enemy territory to protect other primary mission aircraft from enemy fighters en route to and from a target area during offensive missions (i.e., for air interdiction, OCA attack, SEAD, an airborne operation).

(a) Combat Resolution

i. **Protecting Against Enemy Aircraft**

A. Roles: Fighter Sweep = Attacker, CAP = Defender

B. Differential: Attacker DF value - CAP DF Value

C. CRT: DF

ii. **Protecting against ADA (Dynamic SEAD)**

A. Roles: Fighter Sweep = Attacker, ADA = Defender

B. Differential: Attacker SEAD Value - Defender Indirect Fire Value

C. CRT: A2G **(-2 DRM)**

(b) **Fighter Sweep.** Fighter sweep is an OFFENSIVE mission by fighter aircraft to seek out and destroy enemy aircraft or targets of opportunity in a designated area. **(GAME USE: Use Fighter Sweeps to clear out enemy Point Air Defense. This can be done in preparation of an Interdiction attack or just to take try and reduce your enemy's available air sorties by destroying their air units.)**

i. Combat Resolution

A. Roles: Fighter Sweep Unit = Attacker, CAP = Defender

B. Differential: Attacker DF value - Defender DF Value

C. CRT: DF

ii. **Note:** If the Fighter Sweep is used to clear a path for a following Interdiction Mission (e.g. a bomber), after all Fighter Sweep attacks have been resolved, CAP units may remain that were unengaged by the Fighter Sweep or other Escort Fighters. If this is the case, CAP units whose ZOC contain the bomber's targeted hex (infrastructure) may engage the bomber resolving combat using the Direct Fire CRT with the CAP unit as the attacker.

*Design Note: Typically fighter escort and fighter sweep operations immediately precede a ground attack mission as a part of a single strike package. The Air Operations gameplay sequence attempts to capture this sequence through the execution of air-to-air combat first.*

### 6.2.2 Global Precision Attack

> *Global Precision Attack is the ability to hold at risk or strike rapidly and persistently, with a wide range of munitions, any target and to create swift, decisive, and precise effects across multiple domains.*
>
> *-Air Force Doctrine Document (AFDD) 1 (Air Force Basic Doctrine Organization, and Command)*

The Global Precision Attack function takes the form of Counterland Operations. The subelements are Air Interdiction (AI) and Close Air Support (CAS).

#### 6.2.2.1 Counterland Operations

> *Counterland operations are air and space operations against enemy land force capabilities to create effects that achieve joint force commander (JFC) objectives.*
>
> *- AFDD 3-03 (Counterland Operations)*

AFDD 3-03 defines two types of air operations for counterland ops:

1. **Air Interdiction (AI).** AI is an air operation conducted to divert, disrupt, delay, or destroy the enemy's military potential before it can be brought to bear effectively against friendly forces, or to otherwise achieve JFC objectives. AI is conducted at such distance from friendly forces that detailed integration of each air mission with the fire and movement of friendly forces is not required. (i.e. Attack outside of friendly artillery range). **Game Use: Air Interdiction missions have to take place**

**behind enemy lines outside of Artillery Range.**

(a) Combat Resolution

   i. **(Protected)** If ADA is (1) present, (2) not the intended target, and (3) not disabled, the aircraft is engaged by the ADA system. If the aircraft survives the ADA encounter it will then engage its intended target as an unprotected target listed next.

      A. Roles: Aircraft = Attacker, ADA = Defender

      B. Differential: Defender Indirect Fire Value - Aircraft SEAD Value

      C. CRT: S2A

   ii. **Unprotected.** If ADA is not present or is disabled the aircraft freely attacks its intended target.

      A. Roles: Aircraft = Attacker, Target = Defender

      B. Differential: Aircraft A2G value - Targets (DF) combat power or infrastructure defense value

      C. CRT: A2G

2. **Close Air Support (CAS).** CAS is air action by fixed- and rotary-winged aircraft against hostile targets that are in close proximity to friendly forces and which require detailed integration of each air mission with the fire and movement of those forces. In general this is an attack within friendly artillery range.

   (a) Combat Resolution

      i. **(Protected)** If ADA is (1) present, (2) not the intended target, and (3) not disabled, the aircraft is engaged by the ADA system. If the aircraft survives the ADA encounter it will then engage its intended target as an unprotected target listed next.

         A. Roles: Aircraft = Attacker, ADA = Defender

         B. Differential: Defender Indirect Fire Value - Aircraft SEAD Value

         C. CRT: S2A

      ii. **Unprotected.** If ADA is not present or is disabled the aircraft freely attacks its intended target.

         A. Roles: Aircraft = Attacker, Target = Defender

         B. Differential: Aircraft A2G value - Targets (DF) combat power or infrastructure defense value

C. CRT: A2G

# 7.0 Maneuver

> **Joint Fires.** *Fires delivered during the employment of forces from two or more components in coordinated action to produce desired effects in support of a common objective.*
> **Joint Fires Support** *Maneuver is the employment of forces in the operational area through movement in combination with fires to achieve a position of advantage in respect to the enemy*
>
> - JP 3-0 (Joint Operations)

## 7.1 Stacking

There is no stacking of counters or combining Combat Strengths during combat. Units may not move through hexes occupied by any force, friendly or enemy.

## 7.2 Zone of Control (ZOC)

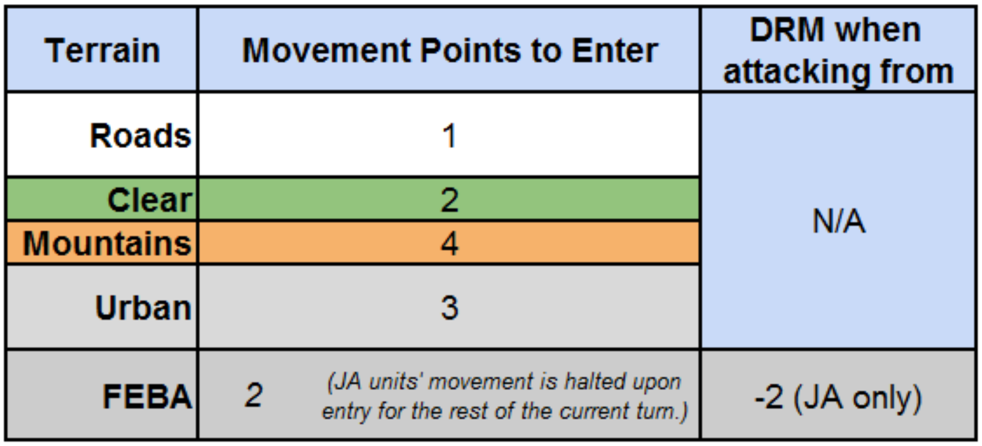
A ZOC exists in the hex occupied by a unit and in each of the six hexes surrounding it. A ZOC affects enemy units for purposes of movement and retreat. Units may not advance after destroying an enemy unit if they are still in another existing enemy unit ZOC.

### 7.2.1 ZOC Effects

1. Units may never enter a hex containing an enemy unit.
2. When entering an enemy ZOC, a moving ground unit must end its movement for the segment.
3. Aircraft movement only affected by an enemy Air Defense unit ZOC and must find a path avoiding ADA ZOCs when conducting interdiction operations. If the path to the intended target passes through a ZOC then the aircraft is engaged by the ADA (use combat resolution instructions under Interdiction: “Protected”).

## 7.3 Terrain

Each unit counter displays that units Movement Points (MP) (see counter legend in 2.2 Counters). Each hex, based on its terrain type, costs a certain number of MP to enter. The Terrain Effects Chart lists the MP cost for all units into each type of hex. NOTE: Roads cost 1 movement point when moving between two hexes with connected roads but cost 2 movement points otherwise.



| Terrain | Movement Points to Enter | DRM when attacking from |
|---|---|---|
| Roads | 1 | N/A |
| Clear | 2 | |
| Mountains | 4 | |
| Urban | 3 | |
| FEBA | 2 *(JA units' movement is halted upon entry for the rest of the current turn.)* | -2 (JA only) |

Terrain Effects Chart (TEC)

### 7.3.1 Installations

Installations are hexes that contain Air Bases (marked by an aircraft symbol), Ports (marked by anchor symbol), and logistics hub. Each installation has a Defense Strength (located on the Game Map) which measures it's survivability against disruption.

#### 7.3.1.1 Installation Occupation

Enemy Installation hexes can be controlled by simple ground movement into the hex.

#### 7.3.1.2 Installation Disruption

Installations can be disrupted through air, missile, and indirect fire attacks.

### 7.3.2 Fortifications

The Korea Barrier System (KBS) is an extensive, in depth, and integrated series of obstacles and barriers, including minefields, concertina wire, and dragon's teeth designed to support the defense of the ROK. It is a significant combat multiplier for the defending CFC forces. The KBS is represented in WARLOC via the FEBA hexes.

#### 7.3.2.1 Barrier Effects

1. All North Korean ground movement is halted upon entering the barrier system, regardless of remaining movement points.
2. All North Korean ground attacks from a unit within the barrier system receive a -2DRM.

## 7.4 Movement Types

### 7.4.1 Ground Maneuver

Ground units move individually from hex to adjacent hex paying the appropriate terrain costs. Units pay movement costs for each hex entered or crossed as detailed on the TEC.

### 7.4.2 Air Maneuver

Originating from the player's Air Base, air units move individually from hex to adjacent hex paying one point per hex.

# 8.0 Combat

## 8.1 Combat Eligibility

Any friendly units that are adjacent to enemy units at the start of a friendly Combat Phase may choose to attack, but are not required to attack. Multiple friendly units may attack the same defending unit but each attack is handled sequentially. All adjacent enemy hexes do not have to be attacked.

## 8.2 Combat Procedure

### 8.2.1 Attack Declaration

The attacking player declares the enemy unit being attacked and identifies their attacking unit. Once an attack is declared, it must be immediately resolved. With air operations, players will declare all their attacks at once then resolve each operation in the sequence indicated under Air Combat in 4.0 Sequence of Play

### 8.2.2 Combat Differential Calculation

In general, the defender's Combat Strength is subtracted from the attacker's Combat Strength.

*Example: The ROK 5th Corps, 3rd Army with a Combat Strength of 7 attacks the North Korean 1st Corps, 1st Operational Echelon with a Combat Strength of 5. The differential calculation is 7 - 5 = 2.*

The only exception to this rule is when ADA defends infrastructure from a TBM. In this case, the JA Commanders SRF unit is the attacker but since the ADA system is attacking the airborne missile, the differential is calculated in the following manner:

Target Infrastructure Value + ADA air defense value - SRF Indirect Fire Value.

### 8.2.3 Combat Results Table Selection

There are four Combat Results Tables (CRT) to choose from:

1. Direct Fire [DF] (Ground to Ground, Air to Air)
2. Indirect Fire [IDF] (Artillery, Surface-to-Surface Missile)
3. Surface to Air [S2A] (Air Defense Artillery Fire on Attacking Aircraft)
4. Air to Ground [A2G] (Air Interdiction, Close Air Support, Suppression of Enemy Air Defense, Attack Operations)

Locate the proper differential column. If the differential is greater than the farthest right column, the attacker uses the farthest right column. If the differential is smaller than the farthest left column, the attacker uses the farthest left column.

### 8.2.4 Determine Dice Roll Modifier (DRM)

A Dice Roll Modifier (DRM) adjusts the total value of the rolled dice. The following cases involve a DRM:

1. Changing an air units mission during the Air Operations - Force Execution Phase (-1 Attack DRM)
2. JA ground unit attacks from a FEBA Hex (-2 Attack DRM)
3. Command Disruption (If C2 node is successfully disabled)
   (a) Seoul (7 Defensive Points) - all ROK ground forces (-1 Attack DRM)
   (b) Pyongyang (7 Defense Points)- Reserve and 2OE forces (-2 Attack DRM)
4. Supply (If logistic node is successfully disabled)
   (a) Daejeon/Nampho (4 Defensive Points) (-1 Defense DRM)

### 8.2.5 Combat Results

1. **Attacker Eliminated (AE)** = Attacker takes a step loss or eliminated; all units involved in the attack take one step loss. If there are no further steps to lose they are destroyed (except bombarding artillery). Defending unit has the option to advance.
2. **Attacker Retreats (AR)** = Unit involved in the attack (except bombarding artillery) are forced to move one hex away from the defender. Defending unit has the option to advance to advance into the hex previously occupied by the attacker.
3. **Equal elimination (EE)** = both attacker and defender damaged or eliminated; If the defender is eliminated and the attacker survives, the attacker may advance into the hex previously occupied by the attacker.
4. **Defender retreats (DR)** = the defending unit is forced to move one hex away from the attacking unit(s). One of the attacking units may advance into the hex previously occupied by the attacker.
5. **Defender eliminated (DE)** = the defending unit is destroyed. One of the attacking units may advance into the hex previously occupied by the attacker.
6. **Disrupt** = Unit/Installation may not attack during the next turn but it may defend. If an ADA system is disrupted by a SEAD attack and then attacked a second time. The Attacker gains a +1DRM.
7. **No Effect (NE)** = Self Explanatory. (Exception: ADA defending against TBM. Refer to 5.1.2 Theater Ballistic Missiles to see how NE is handled in this special case).

## 8.3 Retreat

If the combat resolution results in a retreat, the player must move his unit one hex backwards or sideways from the attacking enemy. The retreating player may not retreat into an enemy ZOC or on top of another unit (friendly or enemy). If the retreating player cannot retreat, the unit incurs a step loss. If the unit is already step-lossed it is destroyed.

# 9.0 Command & Supply

C2 and Logistic Nodes are subject to interdiction. The following are the effects if interdiction attempts are successful and the node is disrupted (Note: effects last for one round total. Example, if the JA Commander successfully interdicts Daejeon, the effects are applied immediately and last until the JA Commander's next turn):

## 9.1 Command Interdiction Effects

1. Seoul (7 Defensive Points): all ROK ground forces -1 Attack DRM
2. Pyongyang (7 Defense Points): Reserve and 2OE forces -2 Attack DRM

## 9.2 Supply Interdiction Effects

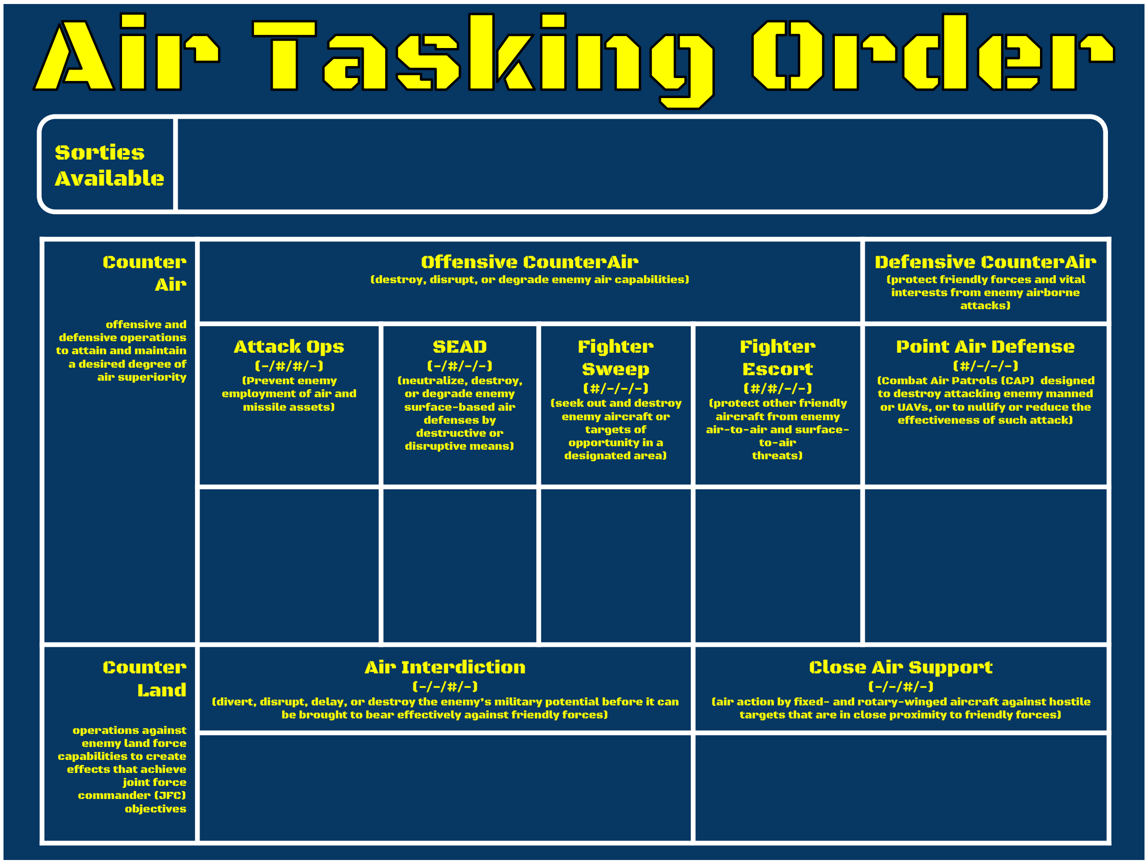
1. Pohang/Wonsan (5 Defensive Points): Reinforcements delayed one turn
2. Daejeon/Nampho (4 Defensive Points): -1 DRM Defense & -1 Movement Point for Ground and Air
3. Osan/Hwangju (5 Defensive Points): 2x Sorties disabled - No movement
4. CrossRoad (3 Defensive Points): 1x additional Ground force unit and 1x additional Sortie disabled

## 9.3 Reinforcement Delay

1. Wonsan
   (a) Move all pending reinforcements for the JA Commander down 1 round on the round tracker found on the game map
2. Pohang
   (a) Pohang is the port in which 3rd Corps arrives in support of Seoul's defense. Each time Pohang is disabled move the 3rd Corps unit counter down one round on the round marker, 3rd Corps will not arrive until this round. If 3rd Corps is delayed past the 5th round, it will arrive at Incheon at the beginning of the CFC Commander's turn on round 6.

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX C:
# WARLOC Air Tasking Orders (ATOs)

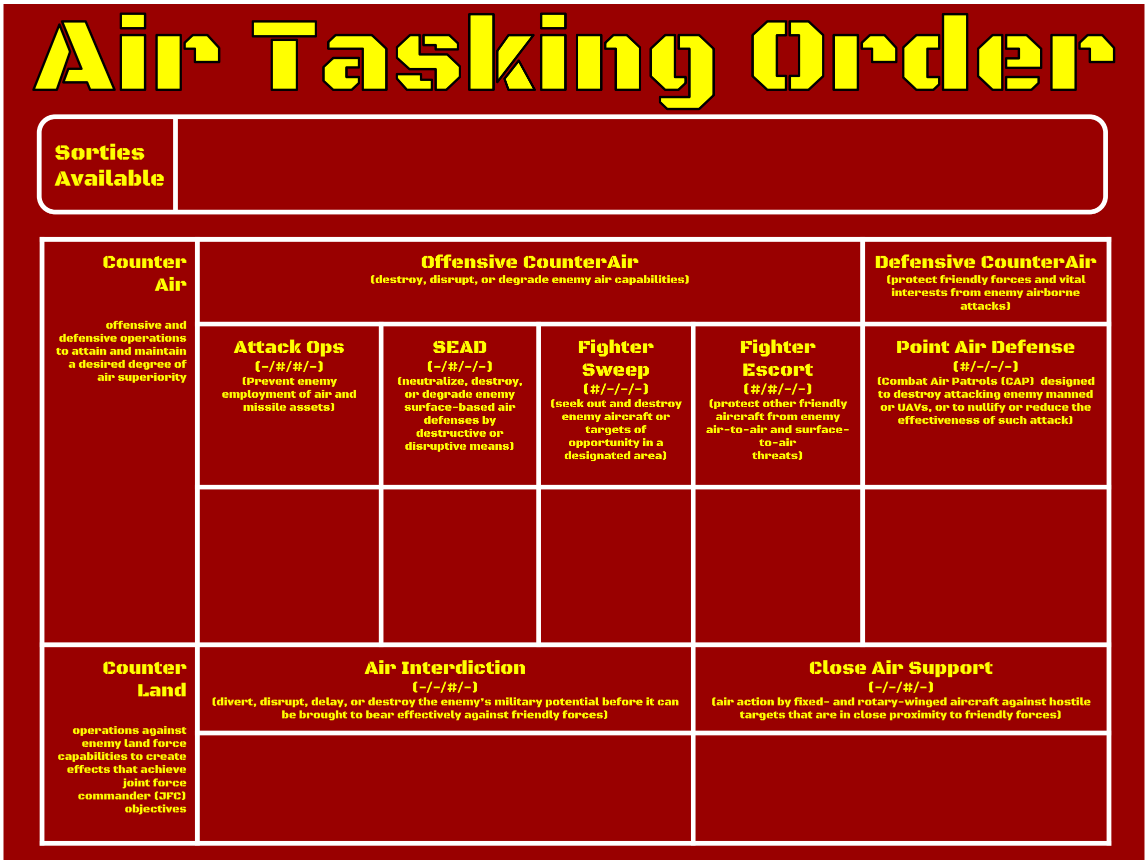


Air Tasking Order

<table>
<tr><td>Sorties Available</td><td></td></tr>
</table>

<table>
<tr><td rowspan="3">Counter Air<br>offensive and defensive operations to attain and maintain a desired degree of air superiority</td><td colspan="4">Offensive CounterAir<br>(destroy, disrupt, or degrade enemy air capabilities)</td><td>Defensive CounterAir<br>(protect friendly forces and vital interests from enemy airborne attacks)</td></tr>
<tr><td>Attack Ops<br>(-/#/#/-)<br>(Prevent enemy employment of air and missile assets)</td><td>SEAD<br>(-/#/-/-)<br>(neutralize, destroy, or degrade enemy surface-based air defenses by destructive or disruptive means)</td><td>Fighter Sweep<br>(#/-/-/-)<br>(seek out and destroy enemy aircraft or targets of opportunity in a designated area)</td><td>Fighter Escort<br>(#/#/-/-)<br>(protect other friendly aircraft from enemy air-to-air and surface-to-air threats)</td><td>Point Air Defense<br>(#/-/-/-)<br>(Combat Air Patrols (CAP) designed to destroy attacking enemy manned or UAVs, or to nullify or reduce the effectiveness of such attack)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">Counter Land<br>operations against enemy land force capabilities to create effects that achieve joint force commander (JFC) objectives</td><td colspan="3">Air Interdiction<br>(-/-/#/-)<br>(divert, disrupt, delay, or destroy the enemy's military potential before it can be brought to bear effectively against friendly forces)</td><td colspan="2">Close Air Support<br>(-/-/#/-)<br>(air action by fixed- and rotary-winged aircraft against hostile targets that are in close proximity to friendly forces)</td></tr>
<tr><td colspan="3"></td><td colspan="2"></td></tr>
</table>

Figure C.1: CFC ATO



Air Tasking Order

Sorties Available

**Counter Air**

offensive and defensive operations to attain and maintain a desired degree of air superiority

**Offensive CounterAir**

(destroy, disrupt, or degrade enemy air capabilities)

**Attack Ops**

(-/#/#/-)

(Prevent enemy employment of air and missile assets)

**SEAD**

(-/#/-/-)

(neutralize, destroy, or degrade enemy surface-based air defenses by destructive or disruptive means)

**Fighter Sweep**

(#/-/-/-)

(seek out and destroy enemy aircraft or targets of opportunity in a designated area)

**Fighter Escort**

(#/#/-/-)

(protect other friendly aircraft from enemy air-to-air and surface-to-air threats)

**Defensive CounterAir**

(protect friendly forces and vital interests from enemy airborne attacks)

**Point Air Defense**

(#/-/-/-)

(Combat Air Patrols (CAP) designed to destroy attacking enemy manned or UAVs, or to nullify or reduce the effectiveness of such attack)

**Counter Land**

operations against enemy land force capabilities to create effects that achieve joint force commander (JFC) objectives

**Air Interdiction**

(-/-/#/-)

(divert, disrupt, delay, or destroy the enemy's military potential before it can be brought to bear effectively against friendly forces)

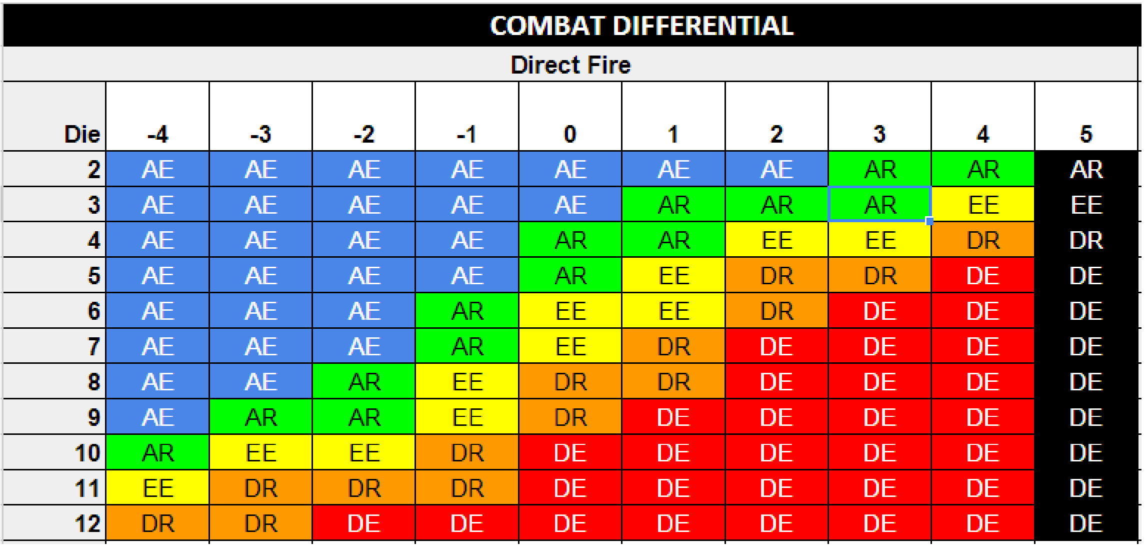
**Close Air Support**

(-/-/#/-)

(air action by fixed- and rotary-winged aircraft against hostile targets that are in close proximity to friendly forces)

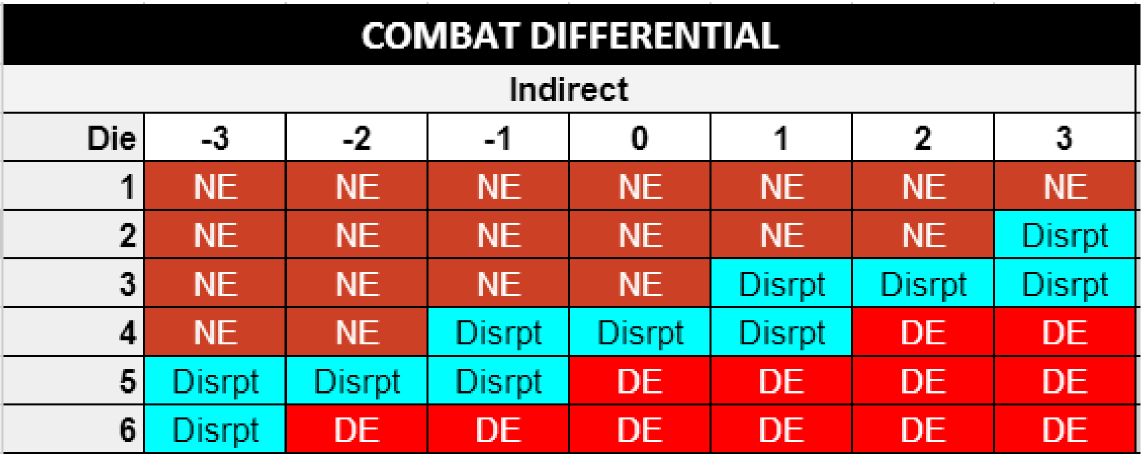
Figure C.2: Juche Alliance ATO

# APPENDIX D: WARLOC Combat Resolution Tables (CRTs)

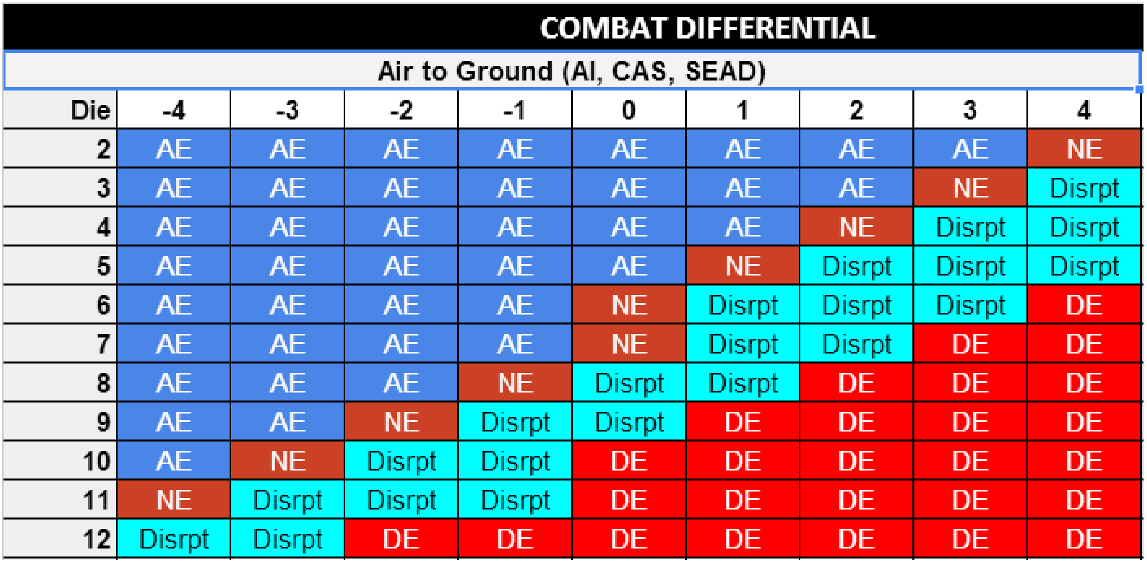


| | COMBAT DIFFERENTIAL | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Direct Fire | | | | | | | | | |
| **Die** | **-4** | **-3** | **-2** | **-1** | **0** | **1** | **2** | **3** | **4** | **5** |
| **2** | AE | AE | AE | AE | AE | AE | AE | AR | AR | AR |
| **3** | AE | AE | AE | AE | AE | AR | AR | AR | EE | EE |
| **4** | AE | AE | AE | AE | AR | AR | EE | EE | DR | DR |
| **5** | AE | AE | AE | AE | AR | EE | DR | DR | DE | DE |
| **6** | AE | AE | AE | AR | EE | EE | DR | DE | DE | DE |
| **7** | AE | AE | AE | AR | EE | DR | DE | DE | DE | DE |
| **8** | AE | AE | AR | EE | DR | DR | DE | DE | DE | DE |
| **9** | AE | AR | AR | EE | DR | DE | DE | DE | DE | DE |
| **10** | AR | EE | EE | DR | DE | DE | DE | DE | DE | DE |
| **11** | EE | DR | DR | DR | DE | DE | DE | DE | DE | DE |
| **12** | DR | DR | DE | DE | DE | DE | DE | DE | DE | DE |

Figure D.1: Direct Fire CRT



| | COMBAT DIFFERENTIAL | | | | | | |
|---|---|---|---|---|---|---|---|
| | Indirect | | | | | | |
| **Die** | **-3** | **-2** | **-1** | **0** | **1** | **2** | **3** |
| **1** | NE | NE | NE | NE | NE | NE | NE |
| **2** | NE | NE | NE | NE | NE | NE | Disrpt |
| **3** | NE | NE | NE | NE | Disrpt | Disrpt | Disrpt |
| **4** | NE | NE | Disrpt | Disrpt | Disrpt | DE | DE |
| **5** | Disrpt | Disrpt | Disrpt | DE | DE | DE | DE |
| **6** | Disrpt | DE | DE | DE | DE | DE | DE |

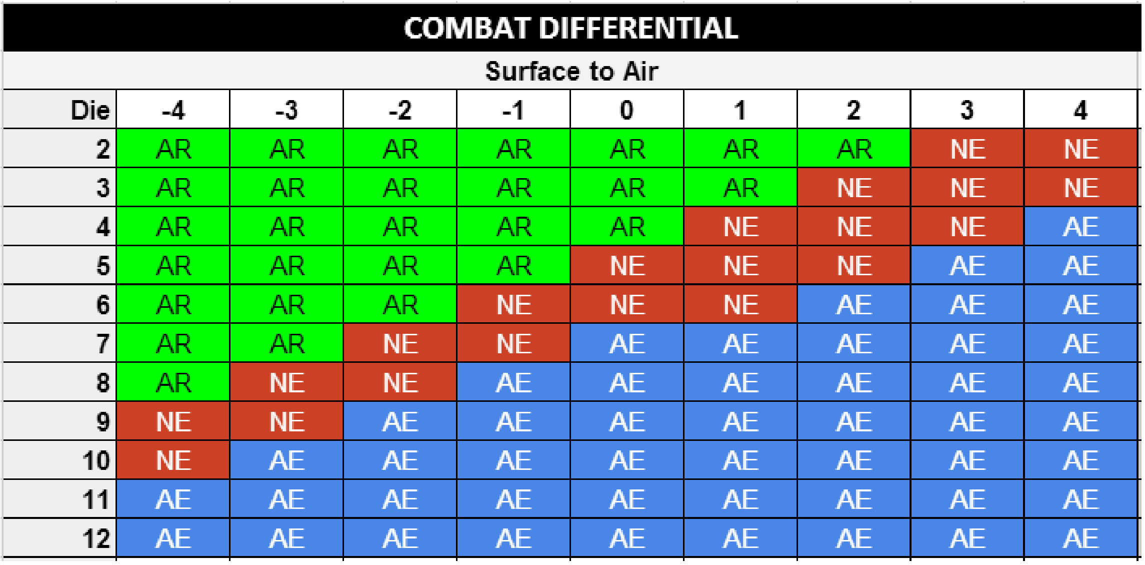


COMBAT DIFFERENTIAL

Air to Ground (AI, CAS, SEAD)

| Die | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|---|---|---|
| 2 | AE | AE | AE | AE | AE | AE | AE | AE | NE |
| 3 | AE | AE | AE | AE | AE | AE | AE | NE | Disrpt |
| 4 | AE | AE | AE | AE | AE | AE | NE | Disrpt | Disrpt |
| 5 | AE | AE | AE | AE | AE | NE | Disrpt | Disrpt | Disrpt |
| 6 | AE | AE | AE | AE | NE | Disrpt | Disrpt | Disrpt | DE |
| 7 | AE | AE | AE | AE | NE | Disrpt | Disrpt | DE | DE |
| 8 | AE | AE | AE | NE | Disrpt | Disrpt | DE | DE | DE |
| 9 | AE | AE | NE | Disrpt | Disrpt | DE | DE | DE | DE |
| 10 | AE | NE | Disrpt | Disrpt | DE | DE | DE | DE | DE |
| 11 | NE | Disrpt | Disrpt | Disrpt | DE | DE | DE | DE | DE |
| 12 | Disrpt | Disrpt | DE | DE | DE | DE | DE | DE | DE |

Figure D.2: Air-To-Ground CRT



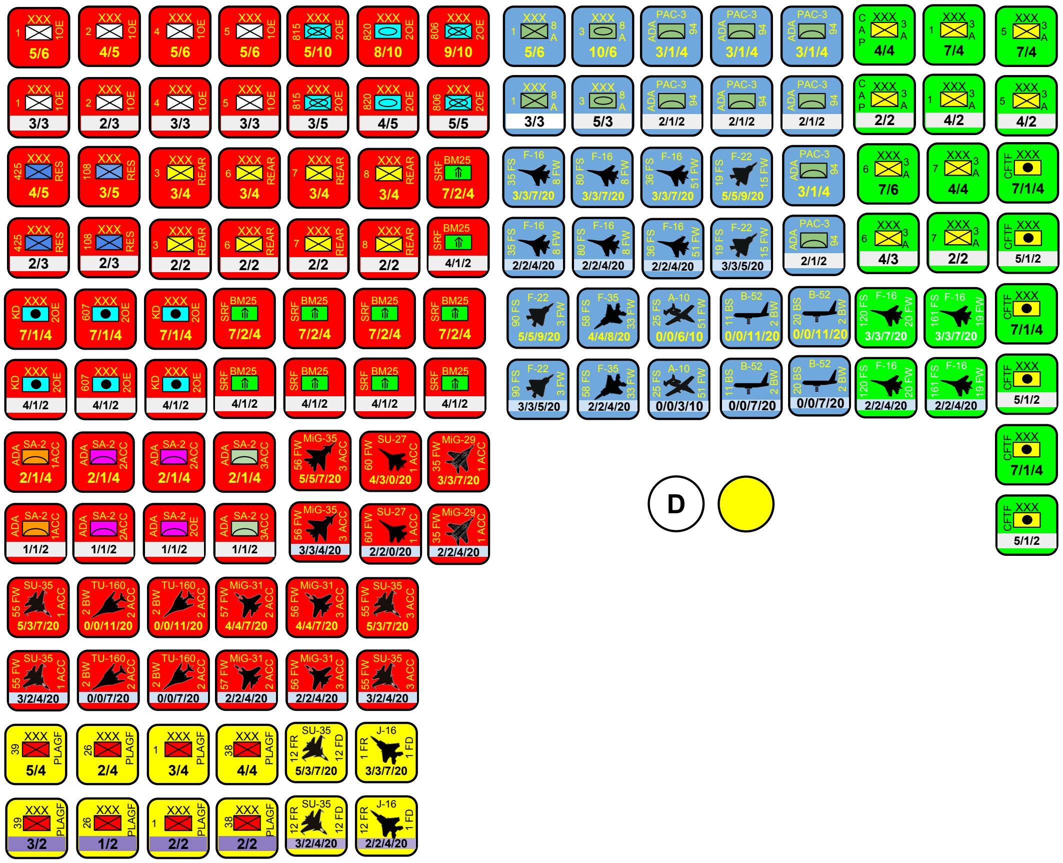
COMBAT DIFFERENTIAL

Surface to Air

| Die | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|---|---|---|
| 2 | AR | AR | AR | AR | AR | AR | AR | NE | NE |
| 3 | AR | AR | AR | AR | AR | AR | NE | NE | NE |
| 4 | AR | AR | AR | AR | AR | NE | NE | NE | AE |
| 5 | AR | AR | AR | AR | NE | NE | NE | AE | AE |
| 6 | AR | AR | AR | NE | NE | NE | AE | AE | AE |
| 7 | AR | AR | NE | NE | AE | AE | AE | AE | AE |
| 8 | AR | NE | NE | AE | AE | AE | AE | AE | AE |
| 9 | NE | NE | AE | AE | AE | AE | AE | AE | AE |
| 10 | NE | AE | AE | AE | AE | AE | AE | AE | AE |
| 11 | AE | AE | AE | AE | AE | AE | AE | AE | AE |
| 12 | AE | AE | AE | AE | AE | AE | AE | AE | AE |

Figure D.3: Surface-To-Air CRT

# APPENDIX E: WARLOC Counters



1 XXX 1OE 5/6 · 2 XXX 1OE 4/5 · 4 XXX 1OE 5/6 · 5 XXX 1OE 5/6 · 815 XXX 2OE 5/10 · 820 XXX 2OE 8/10 · 806 XXX 2OE 9/10 · 1 XXX 8A 5/6 · 3 XXX 8A 10/6 · ADA PAC-3 94 3/1/4 · ADA PAC-3 94 3/1/4 · ADA PAC-3 94 3/1/4 · CAP XXX 3A 4/4 · 1 XXX 3A 7/4 · 5 XXX 3A 7/4
1 XXX 1OE 3/3 · 2 XXX 1OE 2/3 · 4 XXX 1OE 3/3 · 5 XXX 1OE 3/3 · 815 XXX 2OE 3/5 · 820 XXX 2OE 4/5 · 806 XXX 2OE 5/5 · 1 XXX 8A 3/3 · 3 XXX 8A 5/3 · ADA PAC-3 94 2/1/2 · ADA PAC-3 94 2/1/2 · ADA PAC-3 94 2/1/2 · CAP XXX 3A 2/2 · 1 XXX 3A 4/2 · 5 XXX 3A 4/2
425 XXX RES 4/5 · 108 XXX RES 3/5 · 3 XXX REAR 3/4 · 6 XXX REAR 3/4 · 7 XXX REAR 3/4 · 8 XXX REAR 3/4 · SRF BM25 7/2/4 · 35 FS F-16 8 FW 3/3/7/20 · 80 FS F-16 8 FW 3/3/7/20 · 36 FS F-16 51 FW 3/3/7/20 · 19 FS F-22 15 FW 5/5/9/20 · ADA PAC-3 94 3/1/4 · 6 XXX 3A 7/6 · 7 XXX 3A 4/4 · CFTF XXX 7/1/4
425 XXX RES 2/3 · 108 XXX RES 2/3 · 3 XXX REAR 2/2 · 6 XXX REAR 2/2 · 7 XXX REAR 2/2 · 8 XXX REAR 2/2 · SRF BM25 4/1/2 · 35 FS F-16 8 FW 2/2/4/20 · 80 FS F-16 8 FW 2/2/4/20 · 36 FS F-16 51 FW 2/2/4/20 · 19 FS F-22 15 FW 3/3/5/20 · ADA PAC-3 94 2/1/2 · 6 XXX 3A 4/3 · 7 XXX 3A 2/2 · CFTF XXX 5/1/2
KD XXX 2OE 7/1/4 · 607 XXX 2OE 7/1/4 · KD XXX 2OE 7/1/4 · SRF BM25 7/2/4 · SRF BM25 7/2/4 · SRF BM25 7/2/4 · SRF BM25 7/2/4 · 90 FS F-22 3 FW 5/5/9/20 · 58 FS F-35 33 FW 4/4/8/20 · 25 FS A-10 51 FW 0/0/6/10 · 11 BS B-52 2 BW 0/0/11/20 · 20 BS B-52 2 BW 0/0/11/20 · 120 FS F-16 20 FW 3/3/7/20 · 161 FS F-16 19 FW 3/3/7/20 · CFTF XXX 7/1/4
KD XXX 2OE 4/1/2 · 607 XXX 2OE 4/1/2 · KD XXX 2OE 4/1/2 · SRF BM25 4/1/2 · SRF BM25 4/1/2 · SRF BM25 4/1/2 · SRF BM25 4/1/2 · 90 FS F-22 3 FW 3/3/5/20 · 58 FS F-35 33 FW 2/2/4/20 · 25 FS A-10 51 FW 0/0/3/10 · 11 BS B-52 2 BW 0/0/7/20 · 20 BS B-52 2 BW 0/0/7/20 · 120 FS F-16 20 FW 2/2/4/20 · 161 FS F-16 19 FW 2/2/4/20 · CFTF XXX 5/1/2
ADA SA-2 1ACC 2/1/4 · ADA SA-2 2ACC 2/1/4 · ADA SA-2 2OE 2/1/4 · ADA SA-2 3ACC 2/1/4 · 56 FW MiG-35 3 ACC 5/5/7/20 · 60 FW SU-27 1 ACC 4/3/0/20 · 35 FW MiG-29 1 ACC 3/3/7/20 · CFTF XXX 7/1/4
D
ADA SA-2 1ACC 1/1/2 · ADA SA-2 2ACC 1/1/2 · ADA SA-2 2OE 1/1/2 · ADA SA-2 3ACC 1/1/2 · 56 FW MiG-35 3 ACC 3/3/4/20 · 60 FW SU-27 1 ACC 2/2/0/20 · 35 FW MiG-29 1 ACC 2/2/4/20 · CFTF XXX 5/1/2
55 FW SU-35 1 ACC 5/3/7/20 · 2 BW TU-160 2 ACC 0/0/11/20 · 2 BW TU-160 2 ACC 0/0/11/20 · 57 FW MiG-31 2 ACC 4/4/7/20 · 56 FW MiG-31 3 ACC 4/4/7/20 · 55 FW SU-35 3 ACC 5/3/7/20
55 FW SU-35 1 ACC 3/2/4/20 · 2 BW TU-160 2 ACC 0/0/7/20 · 2 BW TU-160 2 ACC 0/0/7/20 · 57 FW MiG-31 2 ACC 2/2/4/20 · 56 FW MiG-31 3 ACC 2/2/4/20 · 55 FW SU-35 3 ACC 3/2/4/20
39 XXX PLAGF 5/4 · 26 XXX PLAGF 2/4 · 1 XXX PLAGF 3/4 · 38 XXX PLAGF 4/4 · 12 FR SU-35 12 FD 5/3/7/20 · 1 FR J-16 1 FD 3/3/7/20
39 XXX PLAGF 3/2 · 26 XXX PLAGF 1/2 · 1 XXX PLAGF 2/2 · 38 XXX PLAGF 2/2 · 12 FR SU-35 12 FD 3/2/4/20 · 12 FR J-16 1 FD 2/2/4/20

Figure E.1: WARLOC Unit Counters

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX F:
# WARLOC Game Map

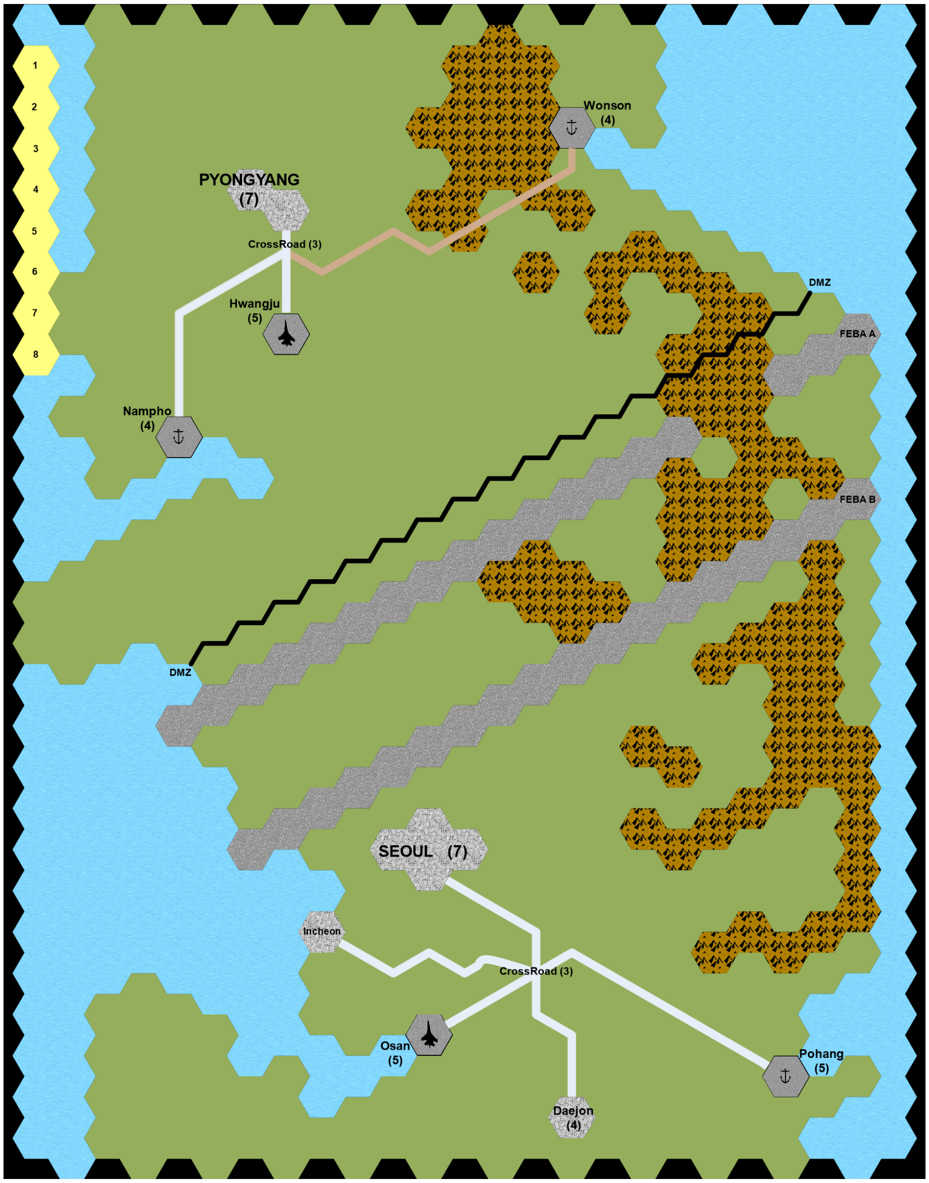


Figure F.1: WARLOC Game Map

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX G:
# Hex Selection Code

Hex Selection function found in WARLOC source code

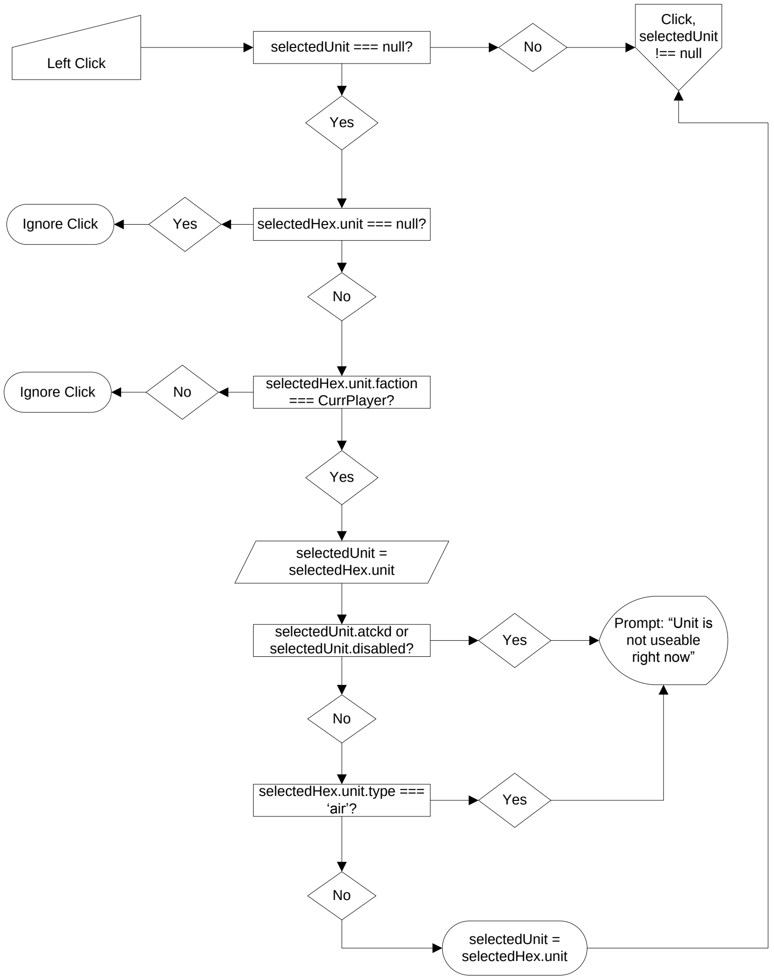
```
/*****************************************************************
*function pickHex                                                *
*Receives canvas coordinates of a mouse click,                   *
 origin of the hex grid and returns the                          *
*column and row index of the hex selected                        *
*Parameters:canvasX,canvasY,canvas height                        *
             hexGridOriginX,hexGridOriginY,half hex side length  *
*****************************************************************/
  var pickHex = function(qx,qy,h,xbb,ybb,u) {
  var hi = 0;                //container for selected hex's column
  var hj = 0;                //container for selected hex's row
  var xm = qx;               //convert canvasX to mathX
  var ym = h - qy;           //convert canvasY to mathY
  var xb = xm - xbb;         //convert mathX to hex grid X coord
  var yb = ym - ybb;         //convert mathX to hex grid Y coord


  var rcol = Math.floor( xb / (3*u) );          //Rectangle Column
  var rrow = Math.floor( yb / (2*sr3*u) );      //Rectangle Row

  /*********************************
  ID the (i,j) coordinates to be
  modified when the region the
  mouse click occurred is identified
  *********************************/
  var LRhi = 2*Math.floor(rcol/2);
  var LRhj = rrow;

  //convert hex grid X coord to rectangle coordX
  var xr = xb - rcol*3*u;
  //convert hex grid Y coord to rectangle coordY
  var yr = yb - rrow*(2*sr3*u) - sr3*u;

  var isEvenCol = (rcol%2===0);          //in an even or odd column?

  if (isEvenCol) {                       //Even Column
    if ( xr > Math.abs(yr)/sr3 ) {
      hi = LRhi;
      hj = LRhj;
    }
    else if (yr>0) {
      hi = LRhi-1;
      hj = LRhj;
    }
    else {
      hi = LRhi-1;
      hj = LRhj-1;
    }
  }
  else {                                  //Odd Column
    if ( xr < u - Math.abs(yr)/sr3 ) {
      hi = LRhi;
      hj = LRhj;
    }
    else if (yr>0) {
      hi = LRhi+1;
      hj = LRhj;
    }
    else {
      hi = LRhi+1;
      hj = LRhj-1;
    }
  }
  return {col:hi, row:hj};         //return the col,row location
}                                  //use with mapHexes[hx.col][hx.row]
```

# APPENDIX H:
# Initial Game Logic Diagrams

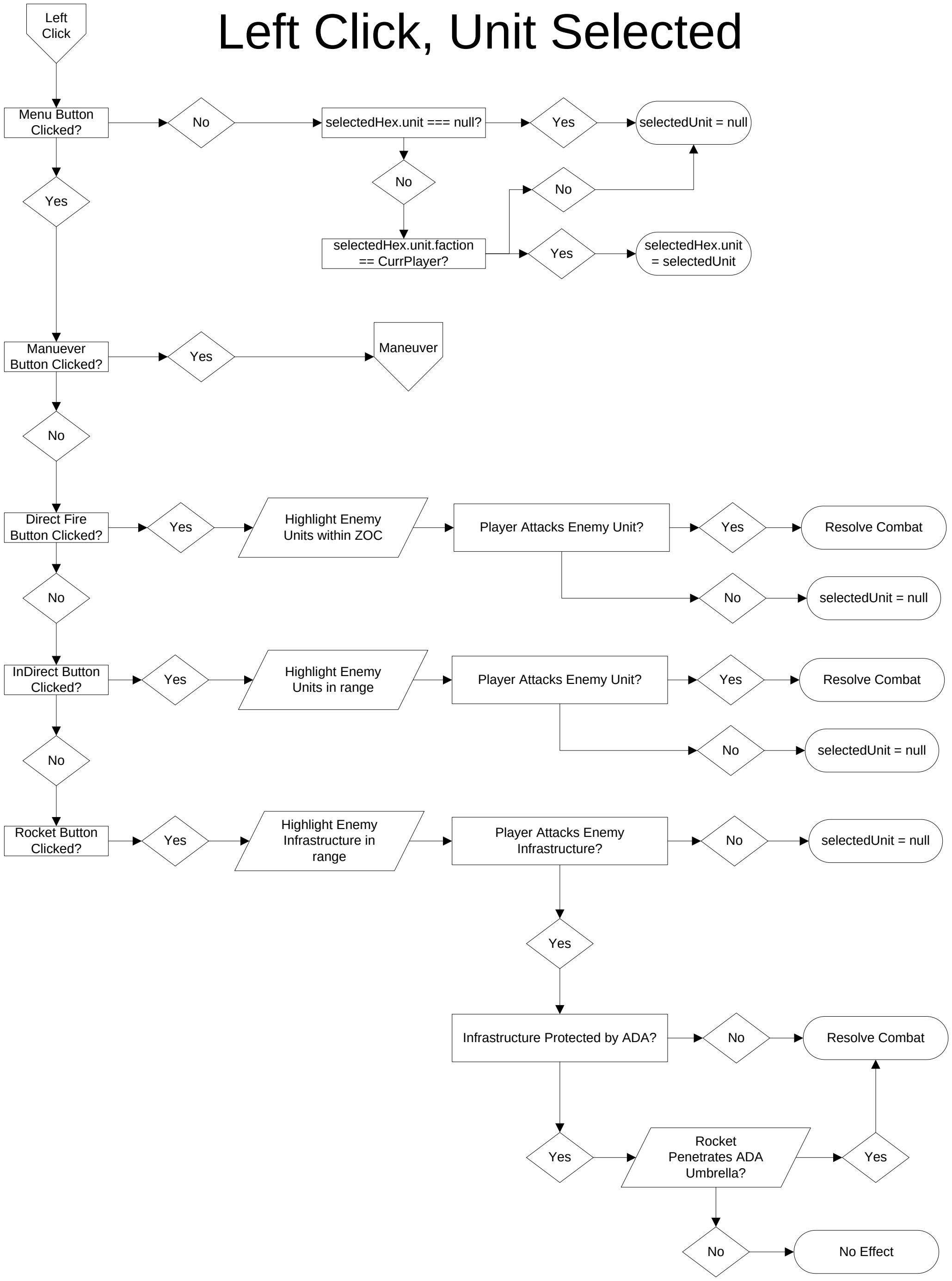
This appendix includes a series of logic diagrams developed to assist in navigating the various actions within WARLOC. Diagrams included

1. Left Mouse Click
2. Left Click, Unit Selected
3. Unit Selected, Maneuver
4. Unit Selected, Direct Fire
5. Calculate DRM
6. Resolve Combat

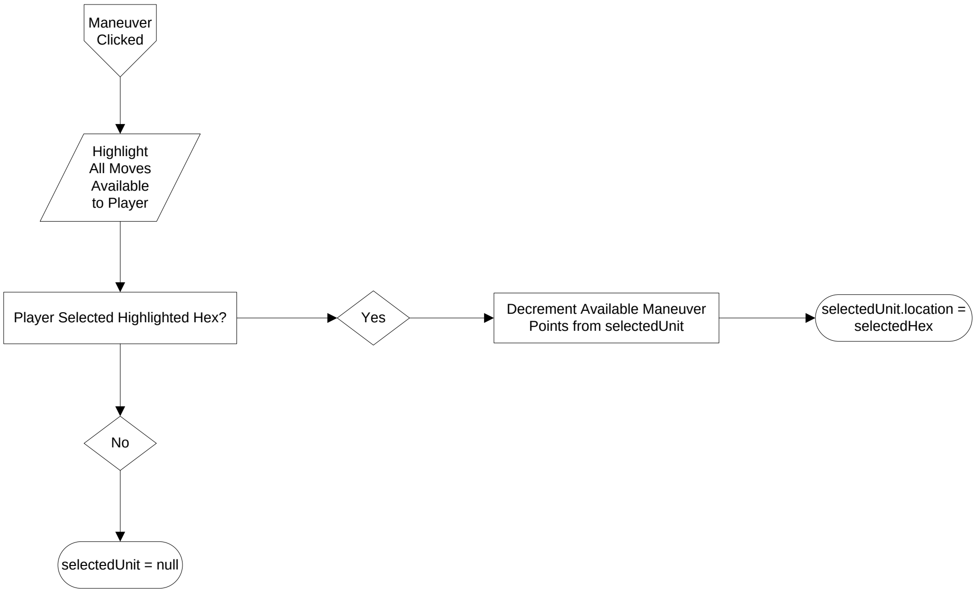
# Left Mouse Click


Left Click
selectedUnit === null?
No
Click, selectedUnit !== null
Yes
Ignore Click
Yes
selectedHex.unit === null?
No
Ignore Click
No
selectedHex.unit.faction === CurrPlayer?
Yes
selectedUnit = selectedHex.unit
selectedUnit.atckd or selectedUnit.disabled?
Yes
Prompt: “Unit is not useable right now”
No
selectedHex.unit.type === ‘air’?
Yes
No
selectedUnit = selectedHex.unit

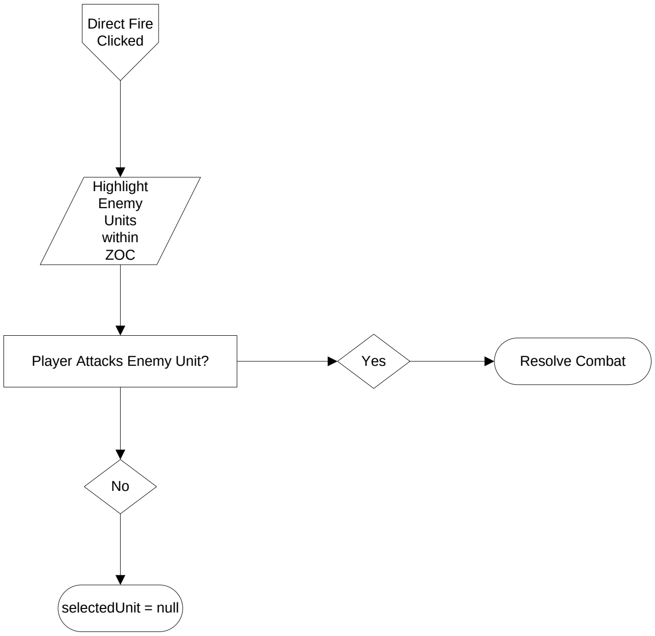
# Left Click, Unit Selected


Left Click
Menu Button Clicked?
No
selectedHex.unit === null?
Yes
selectedUnit = null
No
No
selectedHex.unit.faction == CurrPlayer?
Yes
selectedHex.unit = selectedUnit
Yes
Manuever Button Clicked?
Yes
Maneuver
No
Direct Fire Button Clicked?
Yes
Highlight Enemy Units within ZOC
Player Attacks Enemy Unit?
Yes
Resolve Combat
No
selectedUnit = null
No
InDirect Button Clicked?
Yes
Highlight Enemy Units in range
Player Attacks Enemy Unit?
Yes
Resolve Combat
No
selectedUnit = null
No
Rocket Button Clicked?
Yes
Highlight Enemy Infrastructure in range
Player Attacks Enemy Infrastructure?
No
selectedUnit = null
Yes
Infrastructure Protected by ADA?
No
Resolve Combat
Yes
Rocket Penetrates ADA Umbrella?
Yes
No
No Effect

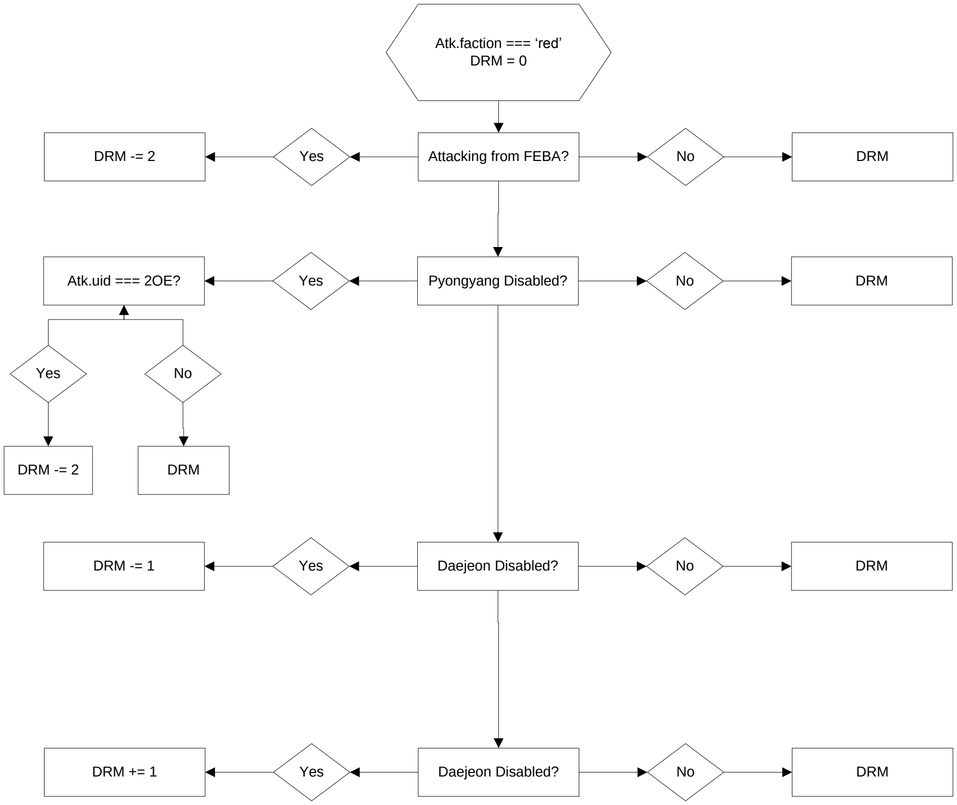
# Unit Selected, Maneuver


Maneuver Clicked
Highlight All Moves Available to Player
Player Selected Highlighted Hex?
Yes
Decrement Available Maneuver Points from selectedUnit
selectedUnit.location = selectedHex
No
selectedUnit = null

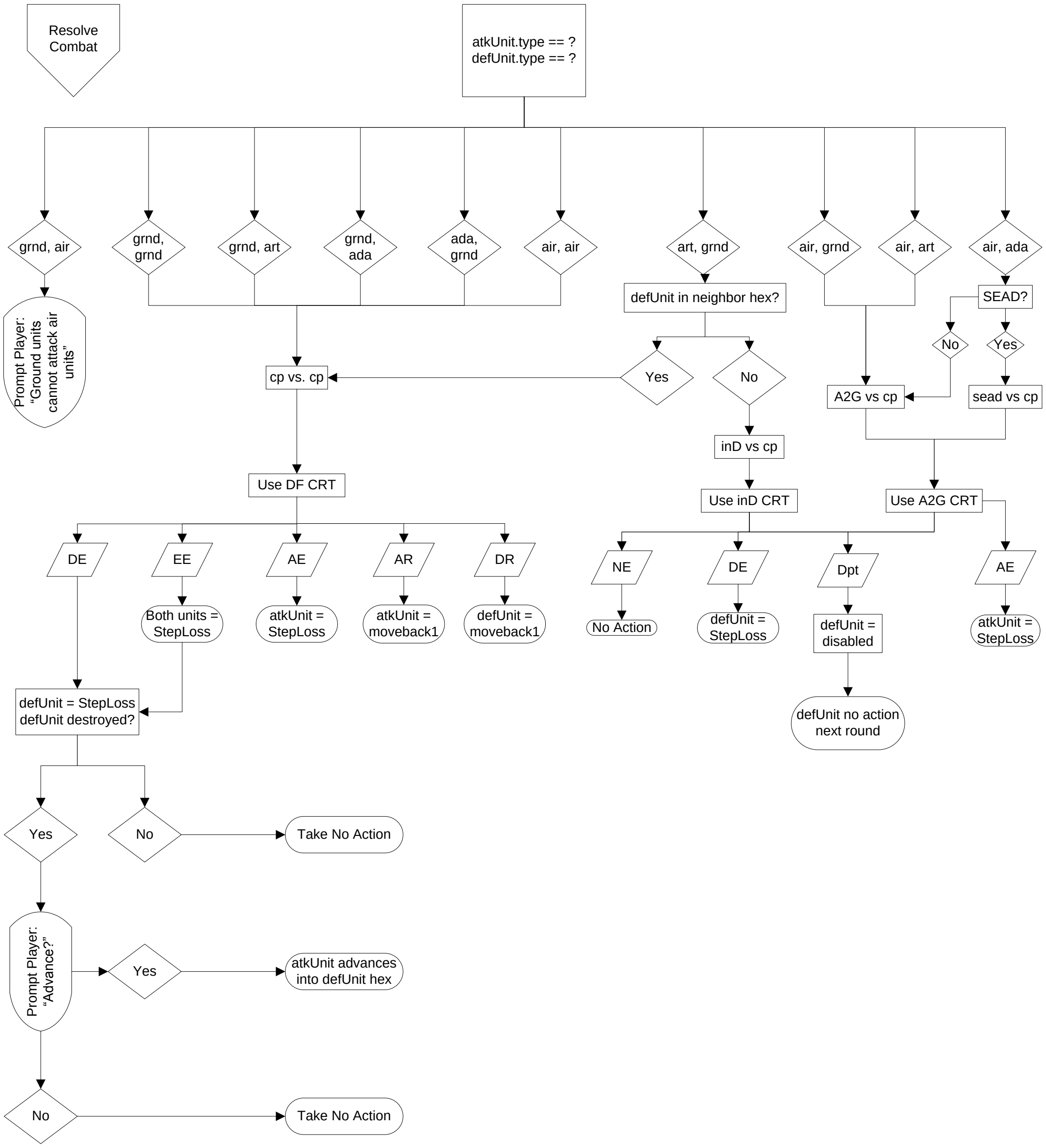
# Unit Selected, Direct Fire



# Calculate DRM



# Resolve Combat



THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX I: detMove Pseudocode

Path finding algorithm in WARLOC source code. Highlights all available moves for a selected unit allowing the player to see all their movement options for the selected unit.

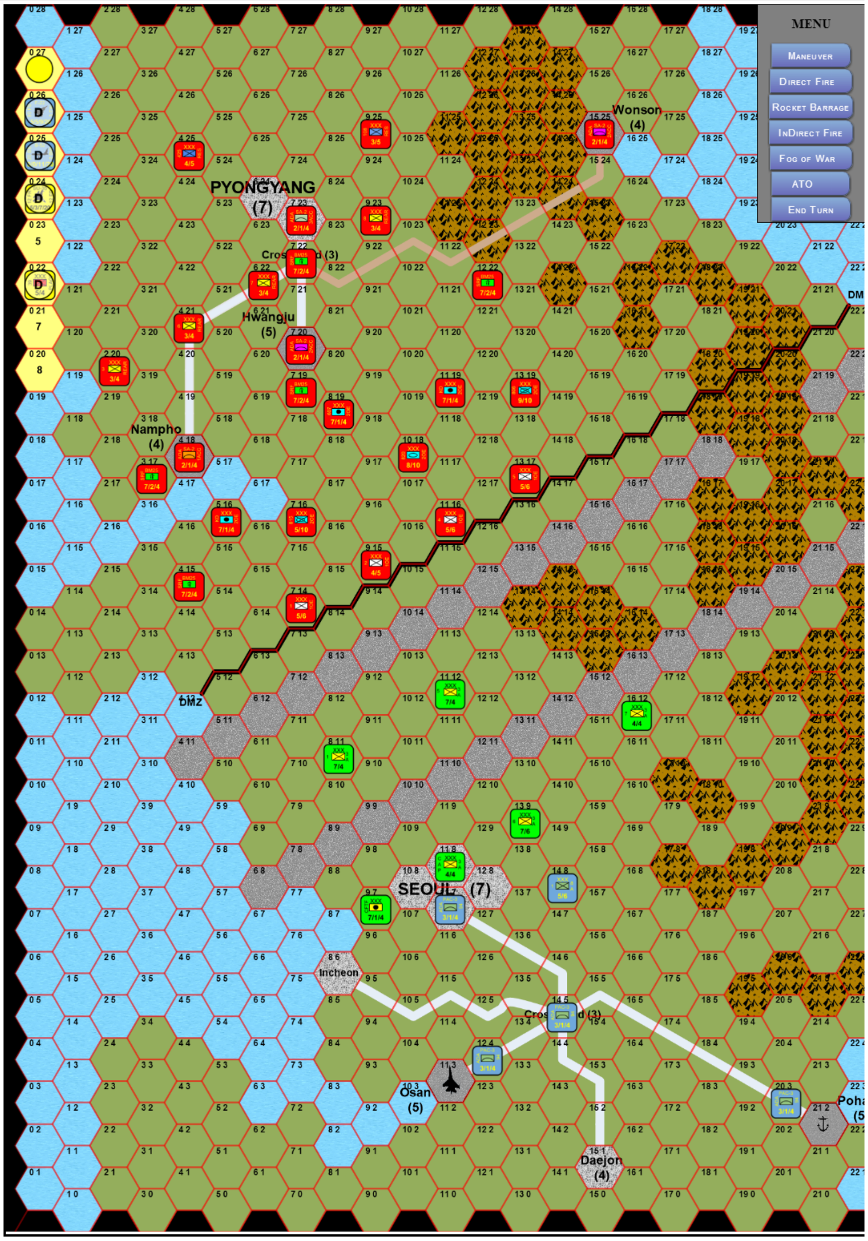
**Arguments**:
*locationHex* : unit's location
*unitMovementPoints* : selected unit's remaining movement points

**Function:**
Determines all available movement options for the selected unit.
Takes into account terrain effects, occupied hexes, and ZOCs
Pushes hexes that the player can move to onto a global array moveOptions[] and visualizes the options to the player by highlighting the hexes

```
Begin Function
  if $locationHex$ is controlled by enemy faction or is 'contested' then
  return Nothing
  end if

  if $unitMovementPoints \leq 0$ then                                    ▷ recursive seed
  return Nothing
  end if

  for all $locationHex$ neighboring hexes do
  $aHex \leftarrow locationHexNeighbor$
  $hxCost \leftarrow 0$
      if $aHex$ is outside selected unit's movement range assuming all roads then skip
      end if
      if $aHex$ terrain is Water then skip
```

```
    end if
    if $aHex$ is Occupied then skip
    end if
    if enemy controls $aHex$ then $hxCost \leftarrow$ Terrain Cost+all remaining unit move pts.
    end if
    if $aHex$ is contested then $hxCost \leftarrow$ Terrain Cost+all remaining unit move pts.
    end if
    if unit faction = $red$ & $aHex$ is FEBA then $hxCost \leftarrow$ 2+all remaining unit move pts.
    end if
    if unit faction = $blue$ & $aHex$ is FEBA then $hxCost \leftarrow 2$
    end if
    if $aHex$ terrain is clear, mountain, urban, or road then $hxCost \leftarrow$ Terrain Cost
    end if
    if $aHex$ is road & $locationHex$ is not road then $hxCost++$
    end if
    if $aHex$ is road & $locationHex$ is urban then treat as connected roads, $hxCost \leftarrow 1$
    end if
    if $hxCost \leq unitMovementPoints$ then highlight aHex & push on moveOptions[]
    end if
                        ▷ Investigate aHex's neighbors with remaining movement points
detMove($aHex, unitMovementPoints - hxCost$)                          ▷ Recursive Call
end for
End Function
```

# APPENDIX J:
# WARLOC Screen Shot



THIS PAGE INTENTIONALLY LEFT BLANK

# List Of References

[1] M. S. Tucker, “Maintaining the combat edge,” *Military Review*, vol. 91, no. 3, pp. 8–16, May/Jun. 2011. [Online]. Available: http://search.proquest.com/docview/869075753?accountid=12702

[2] D. Larsen. (2012, Jun.). Corps-level exercise tests unified land operations capabilities for first time in decade. [Online]. Available: http://www.army.mil/article/82033/Corps_level_exercise_tests_unified_land_operations_capabilities_for_first_time_in_decade/

[3] P. Perla. (2008). So a wargamer and a black swan walk into a bar . . . [Online]. Available: http://www.mors.org/UserFiles/file/PERLA%20-%20Wargamer%20Black%20Swan.pdf

[4] P. Perla, *The Art of Wargaming: A Guide for Professionals and Hobbyists*. Naval Institute Press, 1990, 89028818. [Online]. Available: http://books.google.com/books?id=C_sjPhgaEzwC

[5] K. Kiili, “Digital game-based learning: Towards an experiential gaming model,” *The Internet and Higher Education*, vol. 8, no. 1, pp. 13–24, 2005.

[6] Research@Google. (2013, Nov.). Moore’s law, part1: Brief history of Moore’s law and current state. [Online]. Available: http://googleresearch.blogspot.com/2013/11/moores-law-part-1-brief-history-of.html

[7] D. Flanagan, *JavaScript: The Definitive Guide*, 6th ed., 2011. [Online]. Available: http://my.safaribooksonline.com/book/programming/javascript/9781449393854

[8] (2014). Khan academy: Computer programming. [Online]. Available: http://www.khanacademy.org/computing/cs

[9] HQDA TRADOC, *TRADOC Pam 525-8-2 The U.S. Army Learning Concept for 2015*. Washington, D.C.: Department of the Army, 2011.

[10] K. X. Zhu and Z. Z. Zhou, “Lock-in strategy in software competition: Open-source software vs. proprietary software,” *Information Systems Research*, vol. 23, no. 2, pp. 536–545,598, 06 2012. [Online]. Available: http://www.google.com/url?q=http%3A%2F%2Fmanagement.ucsd.edu%2Ffaculty%2Fdirectory%2Fzhu%2Fpub%2Fdocs%2Fpapers%2FISR_software_competition.pdf&sa=D&sntz=1&usg=AFQjCNH8wyazvmVaQnKgHdUJhX2pg8ljSw

[11] B. Boehm and W. J. Hansen, “Spiral development: experience, principles, and refinements final report,” *STAR*, vol. 38, 2000.

[12] A. Emrich. (2010, may). The rise and fall of wargaming. [Online]. Available: http://wargamecenter.wordpress.com/2010/05/10/the-rise-and-fall-of-wargaming/

[13] U. S. J. C. of Staff, *Joint Publication 1 Doctrine for the Armed Forces of the United States*. Washington, D.C.: Department of Defense, 2013.

[14] (2014). Oxford dictionaries. [Online]. Available: oxforddictionaries.com

[15] C. V. Clausewitz, *On War*. United States: Alfred A. Knopf, 1976.

[16] U. S. J. C. of Staff, *Joint Publication 3-0 Joint Operations*. Washington, D.C.: Department of Defense, 2011.

[17] U. S. J. C. of Staff, *Joint Publication 1-02 Department of Defense Dictionary of Military and Associated Terms (As Amended Through 15 March 2014)*. Washington, D.C.: Department of Defense, 2010.

[18] U. S. J. C. of Staff, *Joint Publication 3-60 Joint Targeting*. Washington D.C.: Department of Defense, 2013.

[19] USAF, *Air Force Doctrine Document 3-0 Operations and Planning*. Washington, DC: Secretary of the Air Force, 2012.

[20] U. S. J. C. of Staff, *Joint Publication 3-01 Countering Air and Missile Threats*. Washington, D.C.: Department of Defense, 2012.

[21] U. S. J. C. of Staff, *Joint Publication 3-03 Joint Interdiction*. Washington, D.C.: Department of Defense, 2011.

[22] U. S. J. C. of Staff, *Joint Publication 4-0 Joint Logistics*. Washington, D.C.: Department of Defense, 2013.

[23] J. Dunnigan. (2000). The complete wargames handbook: How to play, design, and find them. Kindle Book. New York, N.Y.

[24] P. Sabin, *Simulating War: Studtying Conflict Through Simulation Games*. New York, NY: Continuum International Publishing Group, 2012.

[25] J. Peterson, *Playing at the World: A History of Simulating Wars, People and Fantastic Adventures, from Chess to Role-playing Games*. San Diego, CA: Unreason Press, 2012.

[26] R. Smith, “The long history of gaming in military training,” *Simulation and Gaming*, vol. 41, no. 1, pp. 6–19, Feb. 2010.

[27] R. H. Berg, *Wargame Design: The History, Production, and Use of Conflict Simulation Games*. New York: Simulations Publications, Incorporated, 1977, 77081183. [Online]. Available: http://books.google.com/books?id=4QplPQAACAAJ

[28] H. G. Wells. (2003). Little wars and kriegspiel. eBook. [Online]. Available: http://www.gutenberg.org/files/3691/3691-h/3691-h.htm

[29] S. Studio. Battle of the bulge. [Online]. Available: http://www.shenandoah-studio.com/products/botb/

[30] J. Butterfield. (2012, Oct.). Game origins: Design diary #1. [Online]. Available: http://www.shenandoah-studio.com/design-diary-1/

[31] John Tiller Software. (2014). Panzer campaigns. [Online]. Available: http://www.johntillersoftware.com/PanzerCampaigns.html

[32] N. Pavel. (2014). Linuxconsulting: Openpanzer turn based strategy game. [Online]. Available: http://www.linuxconsulting.ro/openpanzer/

[33] A. Harris, *HTML5 For Dummies Quick Reference*. Hoboken, NJ: Wiley Publishing, Inc, 2011. [Online]. Available: http://techbus.safaribooksonline.com/book/web-development/html/9781118078372

[34] (2014). Codecademy. [Online]. Available: http://www.codecademy.com/

[35] (2014). w3schools.com. [Online]. Available: http://www.w3schools.com/

[36] (2014). stackoverflow. [Online]. Available: http://stackoverflow.com/

[37] C. Campbell. (2012, May). Introducing atom.js. [Online]. Available: http://christophercampbell.wordpress.com/2012/05/04/introducing-atom-js/

[38] Easeljs. [Online]. Available: http://www.createjs.com/Docs/EaselJS/modules/EaselJS.html

[39] Gamejs. [Online]. Available: http://gamejs.org/

[40] melonjs. [Online]. Available: http://melonjs.org/

[41] Department of Defense Chief Information Officer. (2009). Clarifying guidance regarding open source software (oss). [Online]. Available: http://dodcio.defense.gov/Portals/0/Documents/OSSFAQ/2009OSS.pdf

[42] S. A. Schearer, “Increasing open source software integration on the department of defense unclassified desktop,” master’s thesis, Naval Postgraduate School, Monterey, CA, Jun. 2008. [Online]. Available: http://handle.dtic.mil/100.2/ADA483625

[43] (2014). Brackets blog. [Online]. Available: http://blog.brackets.io/

[44] (2014). adobe/brackets. [Online]. Available: http://github.com/adobe/brackets/blob/master/LICENSE

[45] (2014). Cloud9 ide. [Online]. Available: http://c9.io/site/company/

[46] J. Loeliger and M. McCullough, *Version Control with Git*, 2nd ed. O’Reilly Media, Inc., 2012. [Online]. Available: http://techbus.safaribooksonline.com/book/databases/content-management-systems/9781449345037

[47] Wikipedia. (2014, May). Github. [Online]. Available: http://en.wikipedia.org/wiki/GitHub

[48] D. Pilone and T. Pilone, *Getting Started: Getting mobile with IOS*, 3rd ed., ser. Head First iPhone and iPad Development. O’Reilly Media, Inc., 2013. [Online]. Available: http://techbus.safaribooksonline.com/9781491950074/meet_suecomma_your_new_boss_html?percentage=0&reader=html

[49] The GIMP Team. (2014, jan). Gimp. [Online]. Available: http://www.gimp.org/

[50] J. Gumster and R. Shimonski, *GIMP Bible*. Indianapolis, IN: John Wiley & Sons, 2010. [Online]. Available: http://techbus.safaribooksonline.com/book/photo-and-graphic-manipulation/9780470523971

[51] J. Uckelman. (2014). Vassal. [Online]. Available: http://www.vassalengine.org/

[52] E. Messina, *VASSAL 3.1 User’s Guide*, 1st ed., 2011. [Online]. Available: http://www.vassalfactory.org/vassalfactory/files/userguide.pdf

[53] J. Chone. (2010, Mar.). Compiled web vs. interpreted web. [Online]. Available: http://www.bitsandbuzz.com/article/compiled-web-vs-interpreted-web/

[54] E. Boyes. (2008, feb). Gdc ’08: Are casual games the future? [Online]. Available: http://www.refworks.com.libproxy.nps.edu/refworks2/default.aspx?r=references;MainLayout::init

[55] J. Schafer. (2013). Turn-based vs real-time. [Online]. Available: oxforddictionaries.com

[56] M. Csikszentmihalyi, *Flow : the psychology of optimal experience*, 1st ed. New York: HarperPerennial, 1991, c1990.

[57] J. McGonigal, *Reality is Broken*. New York: Penguin Press, 2011.

[58] R. S. Baker, S. K. D'Mello, M. T. Rodrigo, and A. C. Graesser, "Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments," *International Journal of Human-Computer Studies*, vol. 68, no. 4, pp. 223 – 241, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581909001797

[59] USAF, *Air Force Basic Doctrine, Organization, and Command*. Washington, DC: Secretary of the Air Force, 2011.

[60] Office of the Secretary of Defense, "Military and security developments involving the democratic people's republic of korea," Office of the Secretary of Defense, Tech. Rep. C-F322D50, 2013. [Online]. Available: http://www.defense.gov/pubs/report_to_congress_on_military_and_security_developments_involving_the_dprk.pdf

[61] S. Winny, *Chapter 12: Serious Games as an Instrument of Non-Formal Learning*, ser. Serious Games and Virtual Worlds in Education, Professional Development, and Healthcare. Ed. Hershey, PA: IGI Global, 2013. [Online]. Available: http://techbus.safaribooksonline.com/book/digital-media/9781466636736

[62] H. Lorayne and J. Lucas, *The Memory Book*, reprint ed. United States of America: Barnes & Noble Books, 1993.

[63] (2007). Cyberwar xxi. [Online]. Available: http://umahexagon.com/CW21IWAlloc.htm

[64] E. Messina, *VASSAL 3.1 Designer's Guide*, 1st ed., 2012. [Online]. Available: http://www.vassalfactory.org/vassalfactory/files/designerguide.pdf

[65] A. Stearns. (2013). Finally! layout in css. [Online]. Available: http://html5devconf.com/archives/oct2013/speakers/alan_stearns.html#session

[66] E. Bidelman. Github:flexbox. [Online]. Available: https://github.com/ebidel/html5demos/blob/master/html5-demos.appspot.com/static/css/flexbox/index.html

[67] (2012). Grains of sand: CSS3 & html5 demos & templates. [Online]. Available: http://cssdemos.tupence.co.uk/button-styling.htm

[68] K. A. Wilson, W. L. Bedwell, E. H. Lazzara, E. Salas, C. Burke, J. L. Estock, K. L. Orvis, and C. Conkey, “Relationships between game attributes and learning outcomes. review and research proposals,” *Simulation and Gaming*, vol. 40, no. 2, pp. 217–266, Apr. 2009.

[69] (2012). Pacificon game expo. [Online]. Available: http://www.pacificongameexpo.com/

[70] R. Blunt, “Does game-based learning work? results from three recent studies,” in *Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, 2007. [Online]. Available: http://patrickdunn.squarespace.com/storage/blunt_game_studies.pdf

[71] Node.js. [Online]. Available: http://nodejs.org/

[72] G. Luger, *Artificial Intelligence Structures and Strategies for Complex Problem Solving*, 5th ed. Essex, England: Pearson Education Limited, 2005.

# Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California